COMPOSING MUSIC BY SELECTION

CONTENT-BASED ALGORITHMIC-ASSISTED AUDIO COMPOSITION

Gilberto Bernardes de Almeida

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in Digital Media — Audiovisual and Interactive Content Creation

Dr. Carlos Guedes, advisor

Dr. Bruce Pennycook, co-advisor

July 2014

Abstract

Musicians today have access to a vast array of multimedia content on personal and public databases, which consequently expands the possibilities for audio processing. Yet, composing with audio samples is still a very time-consuming task. A major reason for the disconnect between the state-of-the-art technology and current compositional practice is the lack of effective browsing methods for selecting, manipulating, and assembling samples for a particular application context.

My dissertation addresses the aforementioned mismatch by proposing an analysis-synthesis framework for assisting musicians in the daunting tasks of selecting and assembling audio signals, namely by incorporating algorithmic music strategies in the processing chain. I address problems raised by the implementation of audio signals in algorithmic composition by creating higher-level descriptions of sound objects, which drastically reduces their audio data representation and yet provides meaningful and highly flexible information. The proposed strategies for audio stream segmentation and description rely on musicological theories, psychoacoustic dissonant models, and content-based audio processing techniques. Using these frameworks, I finally present algorithmic strategies for style imitation and genuine composition that recombine collections of annotated sound objects for a variety of musical contexts from installations to concert

music.

EarGram, the proof-of-concept software I developed as a result of this study, integrates the proposed analysis-synthesis framework in a concatenative sound synthesis system. I also critically analyze some common issues in concatenative sound synthesis and propose the following three solutions that increase user flexibility and control, in particular for creative purposes: (1) meaningful visualizations of the corpus in relation to specific sound typologies; (2) prioritization strategies and/or weights in the unit selection adapted to particular application contexts; and (3) minimization of concatenation discontinuities between audio units by audio processing techniques.

In sum, this dissertation proposes a description scheme for representing sound objects that offers relevant information for the sound-based composer as well as suitable descriptions for automatically modeling the temporal evolution of musical structure. In addition, the sound objects' descriptions are highly flexible and allow the manipulation of audio signals in known computer-aided algorithmic composition strategies linked to symbolic music representations. Ultimately, earGram helps musicians to easily manipulate audio signals in creative contexts—particularly by assisting in and automating a sound mosaic, which allows greater focus on the creative aspects of music making.

Keywords: algorithmic composition, music analysis, recombination, audio synthesis, concatenative sound synthesis.

Resumo

Atualmente os músicos têm acesso a um vasta gama de conteúdo multimédia em bases de dados pessoais e públicas que, consequentemente, expande as possibilidades para o processamento de áudio. No entanto, compor com amostras de áudio é ainda uma tarefa bastante morosa. A razão fundamental para a discrepância entre o estado-da-arte em tecnologia e a prática atual da composição é a falta de métodos de pesquisa eficazes para selecionar, manipular e montar amostras de áudio num contexto de aplicação específico.

A minha tese aborda a divergência anteriormente referida ao propor um sistema de análise e síntese para assistir o compositor nas exigentes tarefas de seleção e montagem de sinais de áudio, nomeadamente por incorporar estratégias de música algorítmica na cadeia de processamento. Eu abordo problemas que resultam da adopção de sinais de áudio na composição algorítmica através da criação de descrições de objetos sonoros de um nível mais alto que a sua codificação digital. Desta forma, reduz-se drasticamente a sua representação de áudio e providencia-se, no entanto, informações relevantes e altamente flexíveis. As estratégias propostas para a segmentação de áudio em objetos sonoros e a sua consequente descrição baseiam-se em teorias musicológicas, em modelos psicoacústicos de dissonância e em técnicas de processamento baseadas no conteúdo de sinais de áudio. Finalmente, apoiando-me em descrições de áudio, apresento estratégias algorítmicas para imitação de estilo musical e composição genuína que recombinam

coleções de objetos sonoros previamente anotados, e que se adaptam a uma série de contextos musicais desde instalações até música de concerto.

EarGram, o software que implementa o modelo que desenvolvi e que valida os conceitos apresentados, integra o sistema proposto para análise e síntese num algoritmo de síntese concatenativa de som. Aqui, também analiso criticamente algumas questões pertinentes e conhecidas da síntese sonora concatenativa e propus as três soluções seguintes, as quais aumentam a flexibilidade de controlo do utilizador em contextos criativos: (1) visualizações representativas do *corpus* em relação a tipologias de som específicas; (2) estratégias de priorização e/ou factores de ponderação na seleção de unidades adaptados a contextos de aplicação específicos; (3) minimização das descontinuidades resultantes da concatenação de unidades de áudio através de técnicas de processamento de áudio.

Em suma, esta tese propõe um esquema de descritores para representar objetos sonoros, que oferece informação relevante ao compositor de música baseada em som, assim como descrições apropriadas para a criação automática de modelos da evolução temporal da estrutura musical. O esquema analítico desenvolvido demonstra também uma grande flexibilidade e permite a manipulação de sinais de áudio em estratégias de composição algorítmica assistida por computador altamente ligadas a representações musicais simbólicas. Em última analise, earGram auxilia os músicos na manipulação de sinais de áudio em contextos criativos—particularmente por assistir e automatizar o processo de montagem de mosaicos sonoros, permitindo-lhes um maior foco nos aspectos criativos da composição musical.

Palavras-chave: composição algorítmica, análise musical, recombinação, síntese de áudio, síntese concatenative de som.

Acknowledgments

It goes without saying that this dissertation is a collaborative work.

My initial words of gratitude go to my supervisors: Carlos Guedes and Bruce Pennycook, whose orientation, endless revisions, criticism, and support were seminal for completing this dissertation. I am deeply indebted to them for their rigor, patience, and encouragement that allowed me to foresee ways to connect computational research to the world of contemporary music. I must also thank both my supervisors for their outstanding hospitality during my visits to New York University Abu Dhabi and University of Texas at Austin.

I want to acknowledge my colleagues and friends at FEUP: George Sioros, Rui Dias, Filipe Lopes, Rui Penha, Gustavo Costa e Rodrigo Carvalho for their continuous support, endless comments, and corrections of several stages of this document.

I would like to express my gratitude to the composers Nuno Peixoto, Rui Dias, and Ricardo Ribeiro, which showed interest and took the time to learn the software I developed—earGram—in order to use it in their compositions. Their comments, questions, and suggestions were quite useful for improving the usability of the system.

I would like to thank Jessica Pabón, Steven Snowden, and June Snowden for their English corrections on the use of terms and grammar.

I want to acknowledge Adriano Monteiro, Adriano Torres Porres, and William Brent for

their more or less direct help in some components of earGram.

I would like to thank Daniela Coimbra for her exemplary modesty, willingness, patient help, and never ending generosity, which provided me the necessary mental force and courage to withstand deadline pressures.

I must also thank David Silva for his kindness and total availability to help me with the formatting and editing of the document.

I must acknowledge Fabien Gouyon for the heavy task of proofreading some chapters of the manuscript and his many helpful remarks.

I had the honor of being supported by a research grant from Fundação para a Ciência e Tecnologia (grant SFRH / BD / 46205 / 2008), without which it would be almost impossible for me to complete this doctoral program.

I also have the privilege of having an excellent group of friends that not only provided me great support in several stages of the research, but also kept an eye on me in order to not lose the motivation to be on stage. During the research period, I would like to highlight some really important artists/friends with whom I had the pleasure to share the stage: Henk van Twillert, Hugo Marinheiro, Fernando Ramos, Isabel Anjo, and Cláudia Marisa.

I would like to thank my sister Marisa Bernardes de Almeida for her support and endless explanations of mathematical concepts.

My greatest gratitude goes to my parents: Joaquim da Silva Almeida and Maria Rosa Moreira Bernardes, for their long-term journey of love and sacrifice for me.

Table of Contents

Abstract	iii
Resumo	v
Acknowledgments	vii
Table of Contents	ix
List of Figures	xiii
List of Tables	xvi
List of Abbreviations	xvii
List of Sound Examples	xix
Contents of the CD-ROM	xxii
Notes on Terminology	xxx

CHAPTER

1 – Introduction	**1**
1.1 - Motivation	4
1.2 – Approach	6
1.3 - Concatenative Sound Synthesis	10
1.3.1 - Technical Overview	11
1.3.1.1 - Analysis	13
1.3.1.2 - Database	13
1.3.1.3 – Unit Selection	14
1.3.1.4 – Synthesis	15
1.4 - Time Scales of Music	15

1.5 - Outline of the Dissertation 17

PART I - ANALYSIS 19

2 - Sound Morphologies: From Pierre Schaeffer and Beyond 20

2.1 - Describing Sound 21

2.1.1 - A Western Musicology Viewpoint: From Note to Noise 22

2.2 – A Schaefferian Approach to Sound Based Theory 24

2.2.1 - Program of Music Research 26

2.2.2 - Morphological Criteria of Musical Perception 27

2.3 - After Schaeffer 30

2.3.1 - Denis Smalley's Spectromorphology 30

2.3.2 - Lasse Thoresen and the Aural Sonology Project 33

2.4 – Critical Review and Summary 36

3 - Computational Segmentation and Description of Sound Objects 40

3.1 – Introduction 40

3.2 – Audio Descriptors 41

3.2.1 - Audio Description Schemes Inspired in Schaeffer's Typo-Morphology 43

3.3 - Identifying Sound Objects Computationally 45

3.3.1 - Onset Detection 46

3.3.2 - Audio Beat Tracking 48

3.4 - A Musician-Friendly Audio Description Scheme 49

3.4.1 - Criteria of Mass 53

3.4.1.1 - Noisiness 53

3.4.1.2 - Pitch 55

3.4.1.3 - Fundamental Bass 56

3.4.1.4 - Spectral Variability 57

3.4.2 - Criteria of Harmonic Timbre 57

3.4.2.1 - Brightness 60

3.4.2.2 - Width 61

3.4.2.3 - Sensory Dissonance 62

3.4.2.4 - Harmonic Pitch Class Profile 63

3.4.3 - Criteria of Dynamics 63

3.4.3.1 - Loudness 64

3.4.3.2 - Dynamic Profile 64
3.5 - Summary 65

4 – Musical Patterns 67
4.1 - Probabilistic Models of Musical Structure 68
4.1.1 - Modeling Elements of Musical Structure 69
4.1.2 - Establishing Musical Progressions Based on Pitch Commonality 71
4.1.3 - Vertical Aggregates of Sound Objects Based on Sensory Dissonance 72
4.2 - Audio Similarity 74
4.3 - Clustering 78
4.3.1 - *K*-means 79
4.3.2 - Quality-Threshold Clustering 80
4.3.3 - Density-Based Clustering 81
4.4 - Visualizations 82
4.4.1 - Sound-Space 83
4.4.1.1 - Two-Dimensional Visualizations of the Corpus Using Binary Sets of Descriptors 84
4.4.1.2 - Multidimensional Reduction of the Descriptor Space 86
4.4.2 - Self-Similarity Matrix 88
4.5 - Mid-Level Description of the Corpus 90
4.5.1 - Meter Induction 90
4.5.2 - Key Induction 92
4.6 – Part I Conclusion 95

PART II - COMPOSITION 97

5 - Organizing Sound 99
5.1 - From Sound to Music: Technical and Conceptual Considerations 100
5.1.1 - Sampling 100
5.1.2 - Micromontage 102
5.1.3 - Granular Synthesis 103
5.1.4 - Musical Applications of Concatenative Sound Synthesis 104
5.1.5 – Sound Structure as a Foundation for Compositional Systems 108
5.1.6 - Music as a Process 110
5.1.7 - Appropriation as a Musical Concept 112

5.2 - Design Strategies for Musical Composition 116
5.3 - Algorithmic Composition 119
5.3.1 – Algorithmic Composition Approaches 120
5.4 - Computational Life Cycle of Music: An Analysis-Synthesis Approach 122

6 - Content-Based Algorithmic-Assisted Audio Composition 125
6.1 - Composing a Corpus of Sound Units 126
6.1.1 - Planning the Macrostructure 129
6.2 – Micro-Time Sonic Design: SpaceMap and SoundscapeMap 131
6.2.1 - SpaceMap 132
6.2.2 - Playing with a Live Input Audio Signal 136
6.2.3 - SoundscapeMap 136
6.3 - Knowledge Engineering: ShuffMeter 140
6.4 - Empirical Induction and Knowledge Engineering: InfiniteMode 144
6.4.1 - StructSeq 144
6.4.2 - ChordSeq 147
6.5 - Synthesis 148
6.6 - Early Experiments and Applications of EarGram in Musical Composition 151

7 - Conclusion 157
7.1 - Summary 158
7.2 - Original Contribution 160
7.3 - Future Work 165

Bibliography 167
Appendices 195
Appendix A - Comparison of Concatenative Sound Synthesis Implementations, Inspired by Schwarz (2006b) and Sturm (2006b) 195
Appendix B - Related Publications 198
Appendix C- Chronological List of Work Created With EarGram 200

List of Figures

1.1 - Overlapping fields addressed by my research, inspired by Ariza (2005). 5

1.2 - Basic building blocks of the musical life cycle computationally modeled in this dissertation. 7

1.3 - Hierarchical organization of the musical time scales considered in the analytical module. 8

1.4 - Algorithmic scheme for CSS. 12

2.1 – Smalley's (1986) morphological profiles of sound objects. 32

2.2 – Smalley's (1986) attack-effluvium continuum. 33

3.1 - Dynamic profile of a sound object and the values extracted from the profile. 65

4.1 – Example of a matrix that exposes the sensory dissonance between all pair of sound objects in the corpus. 73

4.2 – EarGram's parallels coordinates visualization of a corpus comprising a single audio track—*4* by Aphex Twin. 77

4.3 - Visual representation of a corpus of audio units comprising a single audio source—*4* by Aphex Twin—in a 2D-plot whose axes were assigned to the following descriptors: noisiness (*x*-axis) and spectral variability (*y*-axis). The units' color is defined by sensory dissonance, loudness, and duration, by assigning each descriptor to the values of R, G, and B, respectively, and using an additive color model. 83

4.4 - Mapping of an eight-dimensional point to two dimensions. Axes are named as C_x, each dimension of the point as d_{jx}, and P is the final point position (Kandogan, 2000). 88

4.5 - Visualizations of a corpus comprising a single audio track—*4* by Aphex Twin—by a self-similarity matrix (left image) and a related visualization whose color of each cell results from two found clusters in the corpus (rightmost image). The middle image is a detail of the self-similarity matrix, which exposes with detail the color of each cell. 89

4.6 - The Krumhansl and Kessler key profiles for C major and A minor keys (Krumhansl, 1990). 93

5.1 - Relationship between the components of the analysis and composition modules of the framework. 123

6.1 - Three sub-spaces of the corpus defined on top of self-similarity matrix visualization. 130

6.2 - Software interface that allows a user to constrain the corpus in earGram. 130

6.3 - Software interface that allows a user to specify the automation of several parameters of the playing mode spaceMap in earGram through the use of tendency masks. 135

6.4 - Software interface of the playing mode soundscapeMap. 137

6.5 - Comparison between a corpus representation in a 2D-plot before (left image) and after (right image) space optimization. 138

6.6 - Probability distribution given by Clarence Barlow's indispensability formula for the 16 pulses comprising the 16th note level of 4/4, which is defined as the product of prime factors 2x2x2x2. 141

6.7 - Software interface of the playing mode shuffMeter. 142

6.8 - Indispensability weights' distribution for four pulses of a 4/4 bar given by Clarence Barlow's (1987) formula. The three graphs correspond to the clusters depicted in Figure 6.7 and each configuration was scaled and conveys a percentage of variance according to their position on the navigable map. 143

6.9 – Software interface of the playing mode infiniteMode. 146

6.10 - Representation of the amplitude envelope of synthesized units with slight overlap. The yellow box corresponds to the actual duration of the unit, and the red box to the extension added to the unit in order to create the overlapping period. 149

6.11 - Spectrogram representations of the same concatenated output without (top image) and with (bottom image) spectral filtering (expansion-compression). 151

6.12 – Visualization of the corpus that supported the creation of the raw material for the installation *Urban Sonic Impression*. 154

List of Tables

2.1 - Comparison between criteria of music perception of three representative sound-based theories by Pierre Schaeffer, Denis Smalley, and Lasse Thoresen. 38

3.1 - Comparison between computational schemes for the description of perceptual attributes of sound inspired by Schaeffer's typo-morphology. 45

3.2 - Description scheme used to characterize the audio content of sound objects in earGram. 52

3.3 - Perceptual attributes of musical dissonance according to Terhardt. 59

4.1 - Flowchart of the algorithm that reduces the Bark spectrum representation to a single value. 70

List of Abbreviations

List of de abbreviations (ordered alphabetically).

ASP	Aural Sonology Project
CAAC	Computer aided algorithmic Composition
CSS	Concatenative sound synthesis
DBSCAN	Density-based clustering
GRM	*Groupe des Recherches Musicales*
HPCP	Harmonic pitch class profile
IRCAM	*Institut de Recherche et Coordination Acoustique/Musique*
K-S	Krumhansl-Schmuckler
K-K	Krumhansl-Kesseler
MDS	Multidimensional scaling
MIR	Music information retrieval
MIDI	Musical Instruments Digital Interface
MPEG	Motion Pictures Expert Group
PCA	Principal Component Analysis

PROGREMU	Program of music research
PSOLA	Pitch synchronous overlap add
QT-clustering	Quality-threshold clustering
TOM	*Traité des Objects Musicaux* (Treaty of Musical Objects)
TTS	Text-to-speech

List of Sound Examples

1 - Three musical phrases synthesized in spaceMap by navigating the visualization of the corpus with the same trajectory and utilizing the following pairs of audio features: (1) fundamental bass and spectral variability; (2) width and sensory dissonance; and (3) noisiness and loudness, respectively.

2 - Sound generated in soundscapeMap by recombining audio units from a corpus resultant from the onset segmentation of a tropical forest recording. The target was defined by navigating the soundscapeMap's interface, which largely corresponds to a trajectory that goes from sparser and smoother (lower-left corner of the interface) to denser and sharper (top-right corner of the interface).

3 - Sound generated in soundscapeMap by recombining audio units from a corpus resultant from the onset segmentation of a storm recording. The target was defined by navigating the soundscapeMap's interface, which largely corresponds to a trajectory that goes from sparser and smoother (lower-left corner of the interface) to denser and sharper (top-right corner of the interface).

4 - Three phrases generated in sounscapeMap that synthesize targets that change gradually from the most consonant to the most dissonant—all remaining parameters, density-sparsity and smoothness-sharpness, remain unchanged.

5 - Sound example generated in shuffMeter. It exposes the utilization of different instrumental sound clusters extracted from a collection of both drum and bass samples.

6 – Sound generated by recombining and layering different sound clusters extracted from Bob Marley's *Don't Worry, Be Happy*, sung by Bobby McFerrin, utilizing a predefined meter of 4/4.

7 – Sound generated by recombining and layering different sound clusters extracted from Bob Marley's *Don't Worry, Be Happy*, sung by Bobby McFerrin, utilizing a predefined meter of 3/4.

8 - Sound generated in infiniteMode's structSeq that recombines and extends the initial 28 seconds of Jean-Baptiste Lully's *Les Folies d'Espagne* (1672, LWV 48).

9 - Sound generated in infiniteMode's chordSeq that recombines and extends the initial 28 seconds of Jean-Baptiste Lully's *Les Folies d'Espagne* (1672, LWV 48).

10 - Random recombination of audio segments extracted from Jean-Baptiste Lully's *Les Folies d'Espagne* (1672, LWV 48), which have also been used in sound examples 8 and 9.

11 - Electronic composition *Schizophonics* (2012) by Rui Dias.

12 – Scale produced by the organization of sound samples according to the brightness of the audio units from the Porto Sonoro sound bank utilized in *Urban Sonic Impression* (2013).

13 – Scale produced by the organization of sound samples according to the brightness of the audio units from the Porto Sonoro sound bank utilized in *Urban Sonic Impression* (2013).

14 - Excerpt of the sound installation *Urban Sonic Impression* (2013) by Rui Dias and Gilberto Bernardes.

15 - Live recording of Nuno Peixoto's song *Your Feet* (2012), performed by Rita Redshoes (voice) and Nuno Aroso (percussion).

16 - EarGram's "sonic transcription" of the composition *Your Feet* (2012) by Nuno Peixoto synthesized with piano and clarinet sounds (sound example 19).

17 - EarGram's "sonic transcription" of the composition *Your Feet* (2012) by Nuno Peixoto synthesized with piano and clarinet sounds (sound example 19).

18 - EarGram's "sonic transcription" of the composition *Your Feet* (2012) by Nuno Peixoto synthesized with piano and clarinet sounds (sound example 19).

19 - Synthesized MIDI version of Nuno Peixoto's song *Your Feet* (2012) with piano and clarinet.

Contents of the CD-ROM

software

- ReadMeFirst.txt
- earGramv.0.18.pd
- absOverview.pd
- COPYING.txt
- **dependencies**
  - **abs**
    - 2d.tabread-help.pd
    - 2d.tabread.pd
    - 2d.tabwrite-help.pd
    - 2d.tabwrite.pd
    - 2dPlot-help.pd
    - 2dPlot.pd
    - average-help.pd
    - average.pd
    - bagFifo-help.pd
    - bagFifo.pd
    - barGraph-help.pd
    - barGraph.pd
    - beatInduction-help.pd
    - beatInduction.pd
    - bus.input.pd
    - bus.output.pd

bus.pd
bus0.output.pd
collection.pd
colorGrid-help.pd
colorGrid.pd
correlate-help.pd
correlate.pd
covariance-help.pd
covariance.pd
data
    K-K-profiles.txt
    psi-functions.txt
    Temperley-profiles.txt
    vera.ttf
dbscan-help.pd
dbscan.pd
depot-help.pd
depot.pd
featureDisplay.pd
first-help.pd
first.pd
grain.pd
hpcp-help.pd
hpcp.pd
indispenser-help.pd
indispenser.pd
k-means-help.pd
k-means.pd
keyInduction-help.pd
keyInduction.pd
list-bounds-help.pd
list-bounds.pd
list-combine-help.pd
list-combine.pd
list-datasort-help.pd
list-datasort.pd

list-euclid-help.pd
list-euclid.pd
list-maximum-help.pd
list-maximum.pd
list-minimum-help.pd
list-minimum.pd
list-permut-help.pd
list-permute-help.pd
list-permute.pd
list-sliding-help.pd
list-sliding.pd
list-stats-help.pd
list-stats.pd
markovRetrieve-help.pd
markovRetrieve.pd
markovStore-help.pd
markovStore.pd
menu.window.pd
meterInduction-help.pd
meterInduction.pd
minmax-help.pd
minmax.pd
mtx-stats-help.pd
mtx-stats.pd
mtxOptimization-help.pd
mtxOptimization.pd
num2rgb-help.pd
num2rgb.pd
onsetDetect-help.pd
onsetDetect.pd
parallelCoordinatesDef.pd
pitchCommonality-help.pd
pitchCommonality.pd
pitchSalience-help.pd
pitchSalience.pd
pitchShift~-help.pd

pitchShift~.pd
qt-clustering-help.pd
qt-clustering.pd
quantize-help.pd
quantize.pd
randomWalk-help.pd
randomWalk.pd
rgb2pd-help.pd
rgb2pd.pd
rgbAverage.pd
roughness-help.pd
roughness.pd
rSlider-help.pd
rslider.pd
rt.buttons.pd
runningAverage-help.pd
runningAverage.pd
runningCollAverage.pd
runningCovariance-help.pd
runningCovariance.pd
runningStats-help.pd
runningStats.pd
scale-help.pd
scale.pd
similMtxDist-help.pd
similMtxDist.pd
smooth_triple.pd
spaceMapDefH.pd
spaceMapDefV.pd
starCoordinates-help.pd
starCoordinates.pd
starPlot-help.pd
starPlot.pd
stats-help.pd
stats.pd
store.param.f.pd

store.param.l.pd

store.param.s.pd

TIE.6.pd

triggerBox-help.pd

triggerBox.pd

vead~-help.pd

vead~.pd

waveformDisplay-help.pd

waveformDisplay.pd

wavestretch~-help.pd

wavestretch~.pd

externals

soundHack

+spectralcompand-help.pd

+spectralcompand~.pd_darwin

tID

bark-help.pd

bark.pd_darwin

bark~-help.pd

bark~.pd_darwin

barkSpec-help.pd

barkSpec.pd_darwin

nearestPoint-help.pd

nearestPoint.pd_darwin

specCentroid-help.pd

specCentroid.pd_darwin

specCentroid~-help.pd

specCentroid~.pd_darwin

specFlatness-help.pd

specFlatness.pd_darwin

specFlatness~-help.pd

specFlatness~.pd_darwin

specFlux-help.pd

specFlux.pd_darwin

specFlux~-help.pd

specFlux~.pd_darwin

specIrregularity-help.pd
specIrregularity.pd_darwin
specIrregularity~-help.pd
specIrregularity~.pd_darwin
specKurtosis-help.pd
specKurtosis.pd_darwin
specKurtosis~-help.pd
specKurtosis~.pd_darwin
specSpread-help.pd
specSpread.pd_darwin
specSpread~-help.pd
specSpread~.pd_darwin
tabletool-help.pd
tabletool.pd_darwin

fx

bCrush~-help.pd
bCrush~.pd
Chorus~-help.pd
Chorus~.pd
Compand~-help.pd
Compand~.pd
DelayFb~-help.pd
DelayFb~.pd
EQ~-help.pd
EQ~.pd
Flanger~-help.pd
Flanger~.pd
Freeze~-help.pd
Freeze~.pd
Morph~-help.pd
Morph~.pd
Revm2s~-help.pd
Revm2s~.pd
Revs2s~-help.pd
Revs2s~.pd

projectExamples

- ReadMeFirst.txt
- project1_chordProgressionC7.csa
- project2_lullyFolies.csa
- project3_rainforest.csa
- project4_rissetMutations.csa
- project5_techno.csa
- **audio**
  - chordProgressionC7.wav
  - lullyFolies.wav
  - rainforest.wav
  - rissetMutations.wav
  - **techno**
    - drumLoop1.wav
    - drumLoop2.wav
    - drumLoop3.wav
    - synthLoop1.wav
    - synthLoop2.wav

soundExamples

- ReadMeFirst.txt
- soundExample1.mp3
- soundExample2.mp3
- soundExample3.mp3
- soundExample4.mp3
- soundExample5.mp3
- soundExample6.mp3
- soundExample7.mp3
- soundExample8.mp3
- soundExample9.mp3
- soundExample10.mp3
- soundExample11.mp3
- soundExample12.mp3
- soundExample13.mp3
- soundExample14.mp3
- soundExample15.mp3
- soundExample16.mp3

soundExample17.mp3

soundExample18.mp3

soundExample19.mp3

Notes on Terminology

In order to clarify and avoid terminological misconceptions, I will define and constrict the use of three concepts, or pairs of concepts, which will be extensively utilized in this dissertation: (1) computer-aided algorithmic composition; (2) concatenative sound synthesis; and (3) sound object/audio unit.

Computer-aided algorithmic composition is a term coined by Christopher Ariza (2005) that combines two labels—computer-aided composition and generative music—and refers to algorithmic composition strategies mediated by a computer. While computer-aided composition emphasizes the use of a computer in composition, generative algorithms assign the nature of the compositional process to algorithmic strategies. I utilize the term CAAC to pinpoint my focus on algorithmic music strategies that are generally intractable without a high-speed digital computer given the central position of computer usage in the field of algorithmic composition.

Concatenative sound synthesis is a sample-based synthesis technique that I adopted in this dissertation as the technical basis of a devised model. In literature, it is common to find descriptions such as musical mosaicing, musaicing, concatenative sound synthesis, and corpus-based concatenative sound synthesis that explain overlapping (or sometimes

identical) approaches—most enhance idiosyncratic aspects of a particular approach to the technique, yet all adopt a common framework. I use concatenative sound synthesis to address the technique in its broad range, avoiding specifying particularities. For a comprehensive definition of the technique please refer to section 1.3.

Sound object denotes a basic unit of musical structure analogous to the concept of note in traditional Western music approaches (Schaeffer, 1966) and **audio unit** refers to an audio segment with any duration and characteristics manipulated in a concatenative sound synthesis system. Despite their differences, for the purpose of this dissertation, I use the terms sound unit and sound object interchangeably, because I limited the use of audio units to sound objects. While sound objects relate to the conceptual basis of my study, which is greatly attached to musicological literature, audio units are used whenever I focus on a more technical consideration of concatenative sound synthesis, which may encompass audio segments of different structural natures than sound objects.

Chapter 1

Introduction

Music and technology have been closely linked since ancient times. It is even unthinkable to speak and discuss music and its history without considering the technological developments associated with it. Musical instruments like the piano and violin, for instance, are a remarkable result of the collaboration between music and technology. Musical instruments not only constitute major pieces of technological mastery, but are also seminal for the development of musical expression. As Curtis Roads notes, "the evolution of musical expression intertwines with the development of musical instruments" (Roads, 2001, p. 2).

Given the close link between music and technology, it does not seem surprising that the rapid expansion of electronic technology in the late 19th century had a tremendous impact on musical practice a few decades later. In the beginning of the 20th century, the ability to record, amplify, reproduce, and generate sound by electronic means tremendously affected the way we perceive, interpret, and compose music. In the late 1970s, the advent of affordable personal computers offered another avenue for the production of music by electronic means. Computers have become a fundamental tool.

However the music community was, and is still to a certain extent, reluctant to use computers as "creative machines" under the assumption that they are not capable of producing relevant artistic results.

The early days of computer music systems relied almost exclusively in symbolic music representations, in particular the Musical Instrument Digital Interface (MIDI) standard. Symbolic music representations encode physical actions rather than an acoustic reality (Rowe, 2001), and model closely the behavior of a piano keyboard, as well as traditional concepts of music notation (Rowe, 2009). Despite its clean, robust, and discrete representation of musical events, symbolic music codes have many drawbacks. For instance, the MIDI standard, one of the most common symbolic music codes, was recognized since its inception to be slow and very limited in its scope of representation (Moore, 1988).[1]

Audio signals, and in particular digital audio signals, are the most common music representations used today. Contrary to symbolic representations, audio signals encode the music experience, or, in other words, the physical expression or performance. Even if it is a very precise, flexible, and rich representation of the auditory experience and opens up possibilities others than the MIDI or any other symbolic music representation, audio signals also pose crucial problems. Audio signals' low-level representation reclaim the use of algorithmic strategies, importantly including the field of sound and music computing and music information retrieval (MIR), to extract information from the content of the signal.

The field of research concerned with the extraction of information from audio signals is commonly addressed as content-based audio processing, which gained increasing attention in recent years given the large expansion of multimedia content over personal and public databases. Due to the considerable increase of audiovisual contents, it became crucial to develop algorithms for browsing, mining, and retrieving these huge collections

[1] For a comprehensive discussion of symbolic music representation, particularly its limitations please refer to Loy (1985) and Moore (1988).

of multimedia data (Grachten et al., 2009). A substantial body of knowledge has been presented over the last few years, which offers various solutions to help users deal with audio signals in the era of digital mass media production.

The widespread availability of multimedia databases not only affected how users access, search, and retrieve audio, but also enacted critical transformations in how creative industries produce, distribute, and promote music. Research on multimedia information retrieval has also been gradually incorporated in creative work, despite the gap between state-of-the-art research in multimedia information retrieval and usability.

From a creative standpoint, processing audio data is still a very elaborate and time-consuming task. Currently, to create electronic music one usually needs to use software that emulates old analog-tape production means (e.g. audio and MIDI sequencers). These software workstations demand a considerable amount of time to select, segment, and assemble a collection of samples. Despite the large and ever-increasing amount of audio databases, sound-based composers must manage tremendous difficulties in order to actually retrieve the material made available in the databases. One of the most evident and prominent barriers for retrieving audio samples is the lack of appropriate and universal labels for sound description adapted to particular application contexts and user preferences.

In this study, I aim to improve music analysis and composition by devising an analytical framework that describes the audio content of sound objects by minimal, yet meaningful, information for users with a traditional musical education background. Consequently, the audio descriptions will be tested as possible representations of sound objects in computer-aided algorithmic composition strategies (CAAC) greatly attached to symbolic music representations. The ultimate goal is to devise CAAC strategies that deal almost exclusively with audio signals in order to ease the manipulation of audio samples in creative contexts. In addition to the reformulation of known CAAC to process audio

signals, I study new strategies for composing based on the idiosyncrasies of computer music and the description scheme.

The framework proposed will be integrated into an algorithm for concatenative sound synthesis (CSS) and implemented as software (earGram) to test and verify several strategies to analyze and reassemble audio (a detailed description of CSS can be found in section 1.3).

1.1 - Motivation

After completing a Master of Music degree at the Conservatory of Amsterdam, which opened possibilities for aesthetic experimentation with interactive music systems, I had the chance to enroll in a new Doctoral program between two renowned Portuguese Universities—University of Porto and the New University of Lisbon—under the auspices of the University of Texas at Austin.

At first, I was integrated into a project coordinated by my supervisors: "Gestural controller-driven, adaptive, and dynamic music composition systems" (project reference UTAustin/CD/0052/2008). My involvement with the project gave me a solid theoretical and applied knowledge of generative music, which became seminal for fulfilling the objective of this dissertation. By the time I enrolled in the PhD program, I was mainly concerned with the compositional possibilities of using audio signals as the primary music representation in interactive music systems, in particular the use of large collections of audio samples as raw material for musical processing. One of the major reasons motivating my research was the poor sound and expressive qualities of MIDI synthesizers. A major influence is the work of Tristan Jehan, namely his PhD dissertation *Creating Music by Listening* (Jehan, 2005), and soon it became clear that I would work at the intersection of many fields including sound synthesis (namely CSS), algorithmic composition, CAAC, and interactive music (see Figure 1.1).

The ultimate goal of this dissertation is twofold: (1) to reshape the compositional experience of working with audio samples, and (2) to devise an intuitive and intelligible guided search and navigation through large collections of sound-objects oriented towards music composition.

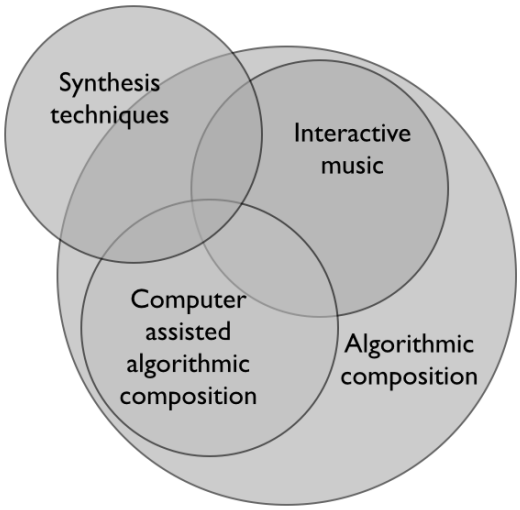


Figure 1.1 - Overlapping fields addressed by my research, inspired by Ariza (2005).

The model I propose aims at reformulating the audio-content description of CSS system audio units through a musical theory and practice standpoint, and targets an audience more familiarized with music theory than with music technology. While my intent is to minimize the usage of computer science terminology, some is unavoidable—particularly concepts related to music information retrieval.[2] In addition, earGram will allow the fast exploration of compositional practices by incorporating several CAAC techniques related to symbolic music representation as unit selection strategies in a CSS system, thus proposing new approaches to explore creatively large collections of audio segments.

[2] Music information retrieval is "a multidisciplinary research endeavor that strives to develop innovative content-based searching schemes, novel interfaces, and evolving networked delivery mechanisms in an effort to make the world's vast store of music accessible to all" (Downie, 2004, p. 12).

1.2 - Approach

In this dissertation I claim the following hypothesis:

> *Sharing the same constitutive elements manipulated through reciprocal operations, morphological and structural analyses of musical audio signals convey a suitable representation for computer-aided algorithmic composition.*

In other words, I suggest that analysis[3] and composition share the same structural elements and can thus be (computationally) seen as complementary operations of a close musical activity cycle. While analysis fragments the sound continuum into constituent elements according to a bottom-up approach in order to reveal and abstract representations of the various hierarchical layers of musical structure, composition elaborates these same elements in an opposite fashion by organizing musical elements from the macrostructure down to the lowest level of musical structure (top-down approach).

The interaction between analysis and composition cannot be discussed without considering music theory. Music analysis and composition not only depart from music theory, but also the constant dialogue between the two fields contributes to music theory with new principles and compositional systems (see Figure 1.2 for an abstract representation the interaction between several agents of the cycle).

Any analysis-synthesis computational approach must describe musical structure. Music theorists have recognized and identified in the temporal span of the music continuum several hierarchical levels (Roads, 2001). The composer's task is undoubtedly to elaborate

[3] Analysis refers to the general process of separating something into its constituent elements and to a certain extent to the examination of the elements or structure of something, typically as a basis for discussion or interpretation. However, it does not imply music analysis, which focuses essentially on the interpretation and elaboration of the elements provided by the analysis carried here.

the several levels of creating a sonic work. Analysis often examines a compositional impulse, while composition often elaborates an analytical impulse.

In order to pursue the aim of this dissertation, I intend to computationally model the music cycle present in Figure 1.2. Specifically, I aim to design a computational system that learns from given musical examples, and/or relies on music theory knowledge, in order to generate meaningful musical results with minimal user interference. The analysis agent encompasses two operations: listening (perception) and learning (cognition), while its complementary agent is composition (action). These two agents are in a constant and reciprocal dialogue with music theory, a repository of knowledge constantly populated with new knowledge generated by the two aforementioned agents.

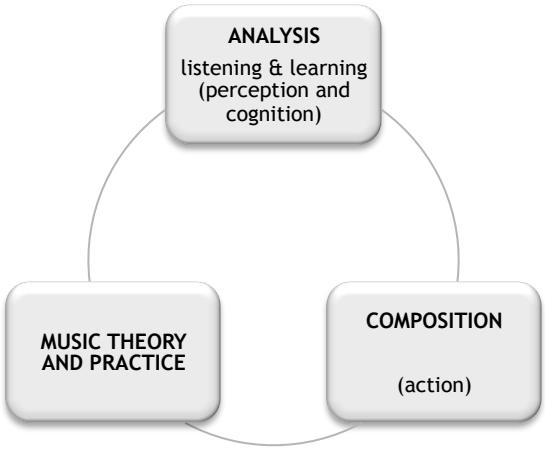


Figure 1.2 - Basic building blocks of the musical life cycle computationally modeled in this dissertation.

My analysis of audio signal content aims at providing representations and revealing patterns of the musical surface higher than the sample temporal unit. In order to do so, I will devise a bottom-up or data-driven computational model for the automatic segmentation and description of sound objects and musical patterns according to criteria of musical perception grounded in sound-based theories by Pierre Schaeffer (1966), Denis Smalley (1986, 1997, 1999), and Lasse Thoresen (2007a, 2007b). Alongside a critical

discussion of the criteria of musical perception proposed in the cited theories, I will devise a set of descriptors for characterizing sound objects; the description scheme is adapted to the idiosyncrasies of a CSS system. Relying on the sound objects' descriptions, I then identify and model higher structural levels of the audio data by grouping sound objects into recognizable patterns up to the macro-temporal level.

Outlined from a music theory and practice standpoint, my model is adapted for music analysis and composition. The outcome of the model intends to provide a rich representation of the audio content in a compact and meaningful representation. However, it does not provide a successful answer to the ultimate goal of the analyst, which is to explain the organization of several events and to reveal how meaning derives from those organizations. Instead, the model provides information that can either allow a different view over the sound material or establish comparisons between vast amounts of material that are not traceable by human senses. A human interference is mandatory in order to determine the causal linkages between the sonic objects and to determine the relationships between patterns (if this level of syntax exists).

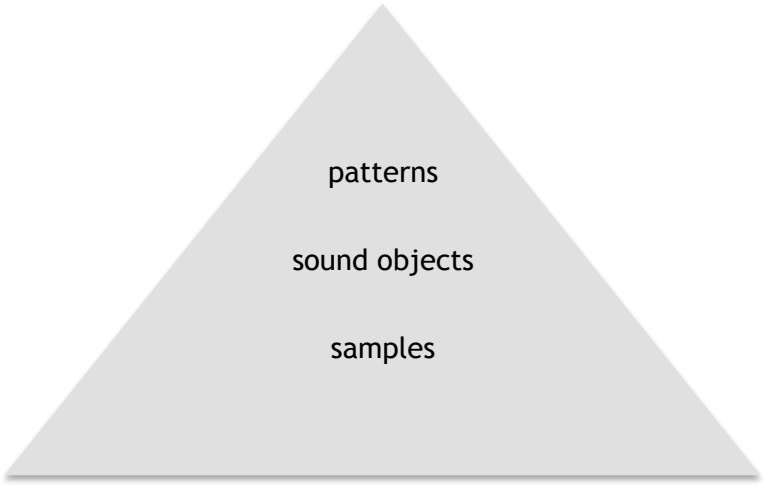


Figure 1.3 – Hierarchical organization of the music time scales considered in the analytical module.

Segmenting the several layers in an audio continuum, along with the description of its constituent units (sound objects), not only provides a groundwork for the analyst, but also for the sound-based composer. In other words, the outcome of my analytical model is suitable for guiding the composition process by reciprocating the analytical operations (i.e. through a top-down or knowledge driven approach). The outcome of my analysis offers the composer a good representation of the audio source's structure and allows a fast and intuitive reorganization of the segments from the macrostructure to the basic element of the musical surface (sound object).

One can compose the macrostructure in earGram by selecting sub-spaces of the corpus that can be assigned to a particular piece, performance, or even to different sections of a work. The process is manual, but guided by several visualizations that expose the structural organization of the corpus, such as similarity matrices and 2D-plots. Some patterns of the audio source(s) structure may also be revealed through the use of clustering techniques in combination with the visualization strategies aforementioned.

The recombination of the sound segments in earGram is automatic and it is mostly done by adapting CAAC algorithms related to symbolic music representations to function as selection procedures in CSS. The CAAC strategies can be guided by music theory knowledge or models created during analysis from user-given examples.

As the name implies, CSS deals with the concatenation or juxtaposition of sound segments, that is, the horizontal dimension of musical structure (e.g. melody, metrical accents, dynamics and properties relating to timbre). However, it is also my intention to expand the CSS scope of action to handle the recombination of units in the vertical dimensions of musical structure (units' simultaneity) as a cause of timbre creation and variance, control of the event density, and (psychoacoustic) dissonance.

Finally, in earGram I will explore the idea that all sonic parameters, such as brightness and sensory dissonance, can be as important as parameters like pitch and duration, which are commonly seen as primary elements of musical structure. I envision all criteria for

sound description as fundamental “building blocks” for compositional systems. This is not to say that every piece composed by these means must use equally all sonic parameters, but that all sonic parameters may be taken into careful consideration when designing a musical work and seen as primary elements of musical structure.

1.3 - Concatenative Sound Synthesis

CSS is “a new approach to creating musical streams by selecting and concatenating source segments from a large audio database using methods from music information retrieval” (Casey, 2009). Briefly, CSS uses a large “corpus” of segmented and descriptor-analyzed sounds snippets, called “units”, and a “unit selection” algorithm that finds the best matching units from the corpus to assemble a “target” phrase according to a similarity measure in the descriptor space.

The first CSS software appeared in 2000 (Schwarz, 2000; Zils & Pachet, 2001) and their technical basis strongly relied on concatenative text-to-speech (TTS) synthesis software—a technique presented in the late 1980s (Schwarz, 2004). CSS began to find its way into musical composition and performance beginning in 2004, in particular through the work of Bob Sturm (2004, 2006b) and Diemo Schwarz (Schwarz, Britton, Cahen, & Goepfer, 2007; this paper documents the first musical compositions and installations exclusively produced by CataRT, a real-time CSS software developed by Schwarz). Currently CSS is considered state-of-the-art in terms of sample-based techniques and content-based audio processing. The technique is at an interesting phase of development and attracts a broad audience of users, researchers, and developers from the scientific to the artistic community. CSS shows great potential for high-level instrument synthesis, resynthesis of audio, interactive explorations of large databases of audio samples, and procedural audio—especially in the context of interactive applications, such as video games. Despite its mature development at engineering and technological levels, CSS is rather undeveloped in terms of aesthetic

and utilitarian concerns. In addition, even though most research in CSS is oriented toward music, the technique lacks substantial contributions in terms of creative output.

In the next section, I will provide an overview of the modules that constitute a CSS system with regard to its technical implementation. Along with the description of the different modules, I will detail the signal data flow of the algorithm and the fundamental terminology associated with each operation. The following overview is restricted to the core components of a CSS and covers the majority of existing CSS software implementations, but it does not target existing variants and subtleties. In addition, the present overview does not distinguish between online or offline approaches. Even if there are some minor differences between the two approaches, the general architecture is the same. Whenever appropriate, I will note the most prominent distinctions.

1.3.1 - Technical Overview

CSS systems commonly comprise four modules: (1) analysis, (2) database, (3) unit selection, and (4) synthesis (see Figure 1.4). Analysis is responsible for segmenting an audio source into short snippets of audio (named units) and describing their content by a set of features (e.g. pitch, loudness, instrument, etc.). The database is responsible for storing all data produced during analysis, which can be later accessed by all of the remaining system components at runtime. The unit selection algorithm is responsible for finding the best matching unit from the database to a target specification. Finally, synthesis converts the output of the unit selection module into audio format. The following paragraphs examine each component of the system individually and point to the respective processing. In addition, the reader can refer to Appendix A for a broad comparison of CSS software according to prominent features such as types of segmentation, audio units' representations, algorithms for unit selection, concatenation type, implementation code, and speed.

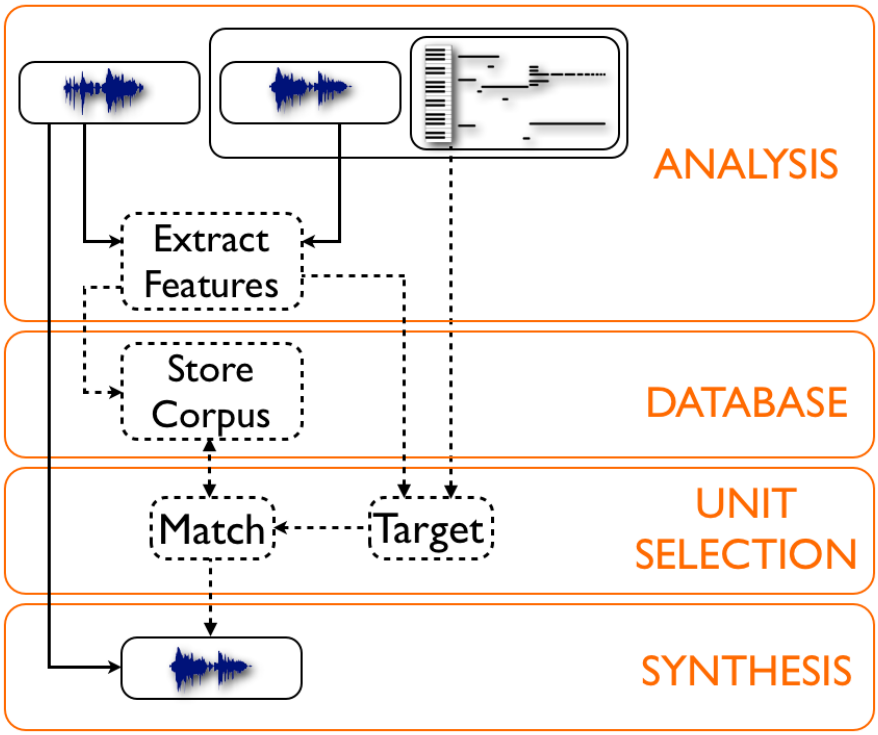


Figure 1.4 - Algorithmic scheme for CSS.

Before describing the algorithm I should clarify two CSS-related concepts and one procedure necessary for the proper functioning of a CSS system. The concepts are unit and corpus, and the procedure is the user-assigned data needed to feed the system. The first tem, unit, is the most basic element of the synthesis algorithm. The algorithm synthesizes new sounds by concatenating selected units that best match a particular target specification. A unit has the same structural value in CSS as a musical note in traditional instrumental music, or even a grain in granular synthesis. Corpus refers to a collection of units. Before any processing takes places, the user must feed a CSS system with audio data that is subsequently segmented into units to form a corpus. This data will be addressed as audio source(s). The choice of the audio source(s) is crucial to the quality of the synthesis because it constitutes the raw material that is concatenated in the end of the processing chain to create new sounds.

1.3.1.1 – Analysis

The analysis module is responsible for two tasks: (1) to segment the audio source(s) into units, and (2) to provide feature vectors that describe the intrinsic characteristics of the units. During segmentation the audio source is divided into several units according to an algorithmic strategy. CSS makes use of different algorithms to segment an audio stream. The outputs are pointers that define the boundaries of each unit.

Analysis comprises a second task that is responsible for extracting relevant audio features from all units in the corpus. The extracted characteristics are further merged into feature vectors that represent the units in all subsequent operations of the system. The feature vectors can be seen as signatures of the units because they provide meaningful and significantly smaller representations of its data. The feature vectors commonly address various characteristics of the units, which, consequently, define a multidimensional space in which the collection of units can be represented.

1.3.1.2 – Database

The database is responsible for storing the data handled and generated during analysis. It includes basic information concerning the units' location in the audio source, along with their representative feature vectors. Several database architectures can be found in CSS software. Most often the database is drawn from scratch in the language in which the application is developed and uses a proprietary analysis data format. Very few implementations adopt common architectures for managing data such as the Structured Query Language (SQL) or its derivatives (Schwarz, 2004; Schwarz, 2006a).

1.3.1.3 - Unit Selection

After creating the corpus, the system is ready to synthesize target phrases. This operation takes place in the two remaining modules of the algorithm, that is, the unit selection and synthesis. Before I detail the unit selection algorithm, I would like to note that it is important to understand the various possibilities of defining target phrases. The target is a representation of a musical phrase, commonly provided by the user, which must be presented to the algorithm as a collection of features in a similar way as the feature vectors created during analysis.

Most systems provide mechanisms for avoiding the user to specify targets as audio features. Instead, what the user commonly presents to the computer is either an audio signal or any other tangible music representation such as MIDI information. Consequently, the system must be able to convert the input representation into a collection of feature vectors. There are two major approaches to the task: (1) data driven and (2) rule-based methods. Data-driven strategies produce targets from the data by applying a set of analytical tools. An example is the transduction of an audio signal into proper feature vector representations. Rule-based approaches encapsulate knowledge to interpret provided information in a meaningful representation to the system. A typical example is the conversion of MIDI data into a string of audio features.

Unit selection encompasses an algorithm that is responsible for searching the corpus and selecting the best matching units according to a given target. In most cases, the unit selection algorithm relies on two conditions to find the best matching unit from the corpus, according to the target specification: the (1) target cost and the (2) concatenation cost. The target cost, also known as local search, is computed by finding the units from the corpus that minimize the distance function to the target in the descriptor space. Different distance metrics are used in CSS, such as Euclidian distance (Hoskinson & Pai, 2001; Hackbarth et al., 2010), Manhattan distance (Brent, 2009), Mahalanobis distance

(Schwarz, 2005), and dot product (Hazel, 2001; Kobayashi, 2003). The concatenation cost, or global search, controls the quality of the units' concatenation. The computation of such criteria may encompass several variables and can be defined by different means. The most common approach is the computation of descriptors' discontinuities introduced by adjacent units. Online systems take care of concatenation cost in a limited way. Most often, online systems consider characteristics of the previously selected unit to constrain the selection processes of the next unit (Schwarz, 2006a). This module outputs the units' numbers or any other pointers that represent the selected units to be synthesized.

1.3.1.4 - Synthesis

Synthesis is the last stage of a CSS system. It is responsible for synthesizing the selected units provided by the unit selection module. The simplest approach to synthesis is to interpret the string of pointers provided by the previous module, find their position on the audio source, and play or render the respective units. The most recent CSS systems also allow the synthesis of overlapping units. This last possibility is quite similar to the playback engine of a granular synthesizer. Some systems also provide the possibility to apply audio effects at the end of the processing chain (Lazier & Cook, 2003; Schwarz et al., 2008).

1.4 - Time Scales of Music

The framework I am proposing deals extensively with the various hierarchical levels of music. In order to establish a set of unambiguous terms to address the various structural levels, I will adopt a taxonomy proposed by Roads (2001) to systematically define all possible time scales of music. Roads' taxonomy presents a comprehensive list of nine music time scales, which cover the various layers of musical structure. The taxonomy

includes the following:

1. “Infinite” refers to a time span of infinite mathematical durations.
2. “Supra” refers to the temporal span beyond the duration of a composition.
3. “Macro” represents the overall architecture, or form, of a musical work.
4. “Meso” comprises the musical phrase formed by dividing the macro structure into smaller parts, or by grouping sound objects into constituent phrase structures of various sizes.
5. “Sound object” is the basic unit of musical structure analogous to the concept of note in traditional instrumental music, but encompassing all perceivable sonic matter.
6. “Micro” encompasses sound particles on a time scale that extends down to the threshold of auditory perception (measured in thousandths of a second or millisecond).
7. “Sample” is the smallest level of the digital representation of an analog audio signal commonly expressed by numerical amplitude values at a fixed time interval.
8. “Subsample” is an unperceivable time scale given its too short duration, and can be understood as the fluctuations that occur below the sample time scale.
9. “Infinitesimal” refers to a time span of infinitely brief mathematical durations.

The four most extreme time scales proposed by Roads—infinitesimal, subsample, infinite and supra, two micro and two macro time scales, respectively—define contextual music situations that are not relevant to the scope of this dissertation because one cannot directly manipulate them in a practical context. However, the reader should be familiar with the five remaining time scales—from sample to macro—because they will be extensively addressed throughout the dissertation.

While defining various time scales of music, Roads provides approximate durations for

all time scales. Even if these values offer a comprehensible suggestion of the duration of each scale of music, they may also provide erroneous information because in a concrete musical work all time scales are identified and definable based on their function, rather than their effective duration.

1.5 - Outline of the Dissertation

I have divided this dissertation into two large parts. Part I (chapters 2, 3, and 4) provides a review of musicological theories for sound description and then offers an algorithmic description scheme for describing and modeling the content of audio signals at various hierarchical levels. Part II (chapters 5, 6, and 7) presents algorithmic strategies to automatically recombine segmented-analyzed audio units, then summarizes the original contribution of this study, and finally provides guidelines for future work.

Chapter 2 presents an overview of three major musicological theories for sound description by Pierre Schaeffer, Denis Smalley, and Lasse Thoresen. In addition, I critically compare the three aforementioned theories with a particular focus on their criteria for the morphological description of sound objects.

Chapter 3 discusses the conceptual and technical considerations that assisted the creation of a description scheme adapted to the automatic characterization of sound objects.

Chapter 4 examines the higher layers of musical structure from an analytical standpoint, focusing on how visualization strategies and statistical analysis help reveal and model musical structure. The visualization strategies are supported by two topics, which are extensively discussed: (1) the computation of similarity between sound objects, and (2) clustering algorithms that help group sound objects that expose similar features and reveal temporal patterns of musical structure.

Chapter 5 provides an overview of the technical and conceptual background of the

framework's generative strategies, particularly an historical perspective of sample-based techniques and compositional systems that contributed to earGram's design. In addition, it is details the articulation between the two major modules of earGram—analysis and composition—and how their interaction establishes a compositional system.

Chapter 6 describes generative music strategies implemented in earGram, from the organization of the macrostructure down to the lowest level of the generated music.

Finally, Chapter 7 discusses my conclusions and original contributions to then provide guidelines for further study.

PART I: ANALYSIS

Ce n'est point avec des idées, mon cher Dégas,

que l'on fait des vers. C'est avec des mots.

— Stéphane Mallarmé

Chapter 2

Sound Morphologies: From Pierre Schaeffer and Beyond

This chapter presents an overview of three representative analytical theories of sound-based works by Pierre Schaeffer (1966), Denis Smalley (1986, 1997) and Lasse Thoresen (2007a, 2007b). Each theory is largely presented according to three topics: (1) methodological premises; (2) conceptual framework; (3) and morphological criteria of sound perception. The first two topics acquaint the reader with select information regarding the foundations and guidelines of the analytical theories necessary in order to then focus special attention on the third topic, laying the groundwork for the development of a computational description scheme presented in the next chapter.

2.1 - Describing Sound

Sound description is an essential task in many disciplines from phonetics and psychoacoustics to musicology and audio processing, which address it for a variety of purposes and through very distinct perspectives. The two disciplinary approaches to sound description most relevant to this dissertation are computational and musicological. Computational refers to content-based audio processing strategies, namely the use of audio descriptors to provide an automatic characterization of an audio signal's content. For example, a computer can easily describe how bright, loud, and stable a sound is by inspecting characteristic present in its digital signal representation.

Content-based audio processing systems that extensively use audio descriptors tend to exclude the analytical operations of the system from the interface. The computational descriptions in content-based audio processing systems like Shazam[4] and Moodagent[5] take place during the implementation, or training, phase of the algorithm and are hidden from the system's interface, thus preventing the user from accessing them. Contrarily, creative applications like Echo Nest Remix API[6] and CataRT (Schwarz, 2006a) give access to the generated audio descriptions and even allow their manipulation, because it is an inherent process of music creation. However, most audio descriptors extracted computationally, like spectral kurtosis and spectral flatness, are not adapted to the terminology of musical practice and are meaningless to musicians.

The gap between computational descriptions of sound and music practice/theory is essentially a problem of terminology, because even if many audio descriptors measure musical or perceptual properties of sound, they are always addressed according to the mathematical operation involved. Developing a set of descriptors adapted to current music practice will increase the usability for those musicians more familiar with music

[4] http://www.shazam.com.
[5] http://www.moodagent.com.
[6] http://echonest.github.io/remix.

theory and practice than with music technology. By unpacking the language, the usability of content-based audio systems would increase considerably, and appeal to a larger audience, most-importantly including musicians. Indebted to previous research by Ricard (2004), Peteers and Deruty (2008), and Schnell, Cifuentes, and Lambert (2010) (each of whom was inspired by Pierre Schaeffer's typo-morphology), the strategy I apply here will offer a description scheme adapted to the needs and knowledge of musicians.

I will now discuss musicological approaches to sound description for two reasons: (1) to present a theoretical basis of the mechanisms behind sound description; and (2) to provide a succinct set of descriptors adapted to music imperatives, particularly composition. In addition, utilizing perceptual criteria like mass, harmonic timbre, and dynamics, based on musicological literature describes abstract sounds independent of their sources because they rely on perceptual characteristics of the audio signal disregarding causal relationships.

2.1.1 - A Western Musicology Viewpoint: From Note to Noise

The emergence of electroacoustic music in the 1940s extended significantly the practice of music creation with new instruments, a myriad of tools resultant from the possibility of recording and diffusing audio, raw material that has been unexplored so far in music composition, etc. Until then, music composition was confined to acoustic instrumental and vocal models and uniquely focused on the manipulation of the following four elements: pitch structures (melody, harmony, aggregates, etc.), rhythm (meter), timbre (restricted almost exclusively to orchestration), and form (theme, motives, macro-form, etc.) (Thoresen, 2007b). These musical elements convey a clear understanding of musical structures that are highly tied to the concept of musical note. The musical note, as the basic unit of composition, favors pitch and duration as primary musical elements over timbre or other attributes of sound.

The appearance of new electronic instruments and sound manipulation strategies broke the paradigm linking sound to the physical object producing it, and allowed composers to work with dimensions that were previously inaccessible or totally disregarded in music composition, particularly the use of all sonic phenomena as raw material for a composition or expanding the act of composing to the sound material itself. In electroacoustic music, the basic structural unit of the composition is no longer the musical note. Instead, the concept of sound object comes to the fore, significantly extending the spectrum of possibilities (from note to noise) without indicating a priori sources or known causes. Electroacoustic music opened the exploration of timbre and reformulated the notion of spectrum as a compositional strategy.

As a result, much electroacoustic music was particularly resistant to traditional analysis and categorization. In addition, the new dimensions explored in electroacoustic music existed for some decades without any theoretical ground or formal definition that could articulate the relevant shift within musical composition. Clearly, a unique set of terms and concepts was needed to discuss, analyze, and interpret electroacoustic music (Smalley, 1986).

In the early years of electroacoustic music theory, the discourse was largely monopolized by engineering terminology, consequently lacking theoretical and aesthetic reflection. In 1966, Pierre Schaeffer presented *Traité des Objets Musicaux* (TOM)—the first substantial treatise on the subject, which addresses the correlation between the world of acoustics and engineering with that of the listener and musical practice. While the technology used by Schaeffer is now outdated, and his argument far from the model presented here, his overall perspective in TOM is valid because of the approach taken to listening and the new concepts and taxonomies of timbre and sound description.

2.2 - A Schaefferian Approach to Sound Based Theory

TOM was the first major essay that attempted to understand and devise an analytical

theory for sound-based works. In TOM, Schaeffer outlines a Program of Music Research (PROGREMU) that provides several stages of action like the definition of different types of sounds along with their morphological description, characterization and organization. These stages aim to abstract musical value from audio signals for particular musical contexts (Landy, 2007). Although it provides a solid foundation for musical composition, TOM is "situated rather in the area of hearing than making, it is descriptive rather than being operational" (Chion, 1983, p. 98). Schaeffer reframes the act of listening to sound by articulating a phenomenological theory that is primarily concerned with the abstracted characteristics of sounds, rather than their sources and causes (Chion, 1983). The theory articulates modes of listening to sound that ultimately establish the basis of a solfeggio for sound-based works.

In order to acquaint the reader the basis of TOM's methodology, four concepts coined by Schaeffer—(1) concrete music (*musique concrète*), (2) listening functions, (3) reduced listening, and (4) sound object— will be examined next. Their order reflects a top-bottom organization of music practice/theory and human perception principles.

According to Schaeffer, concrete music denotes the music created by a group of composers working at the French Radio, which later became the *Groupe des Recherches Musicales* (GRM). Schaeffer provides an explanation of this term, as many others, by referring to binary and antonym concepts (Landy, 2007). The term "concrete" is used to represent a musical reality, a new creation paradigm that opposes abstract music, which was the prevailing composition model for vocal or acoustic instruments at the time. It emphasizes that the raw material for a composition is based on pre-recorded sounds. In contrast, traditional Western music composers start with an abstract idea of the work, which only later achieves its concrete form when performed.

Schaeffer's theory derived from particular active listening functions. He describes four listening functions related to different ways of perceiving and understanding sound: *écouter*, *comprendre*, *entendre* and *ouir*. John Dack and Christine North translated them

to English as: to listen, to comprehend, to hear, and to perceive, respectively (Chion, 1983). To listen refers to the identification of the sound-producing event through the sound. In this case, sound is seen as an index of an event. To comprehend implies the identification of a message transmitted by the sound. This listening function is well illustrated by speech, in which sound is only a "vehicle" carrying meaning for words. To hear is to perceive the intrinsic properties of a sound. These qualities allow us to distinguish between different instruments, for instance. To perceive refers to the discernment of the raw-sound data with no intention of interpreting or qualifying it. It is the lowest level of our auditory perception and can be seen as a kind of passive listening (Chion, 1983).

Another key concept in TOM is reduced listening, which is a listening attitude that focuses on the morphological qualities of sound rather than its causes or meaning. Reduced listening neglects the phenomenon of source identification that is highly linked to vocal and instrumental music and describes Schaeffer's methodological approach adopted in TOM for analyzing the qualities expressed by sound events. As Jean-Claude Risset notes "in the first instance, Schaeffer placed the accent on the primacy of the listening experience and on the necessity to develop a solfeggio of effects as opposed to causes" (as cited in Thomas, 1999, p. 37). It also would be difficult to approach sound with reduced listening as a strategy without available technology for sound recording and diffusion, because we must be exposed to a sound many times to fully grasp its morphology. The repetition of the physical signal prompts more awareness of its perceptual attributes and relegates the listener's attention on the sound source to a secondary level (Schaeffer, 1966). Schaeffer's reduced listening denotes an attitude toward listening that is characteristic of its time: to listen to sounds whose cause is invisible, such as radio broadcasts, telephone conversations, or recorded sounds. The focus on the sonic matter is a guiding principle for the compositional approach of the composers associated with the GRM.

Lastly, let us address the concept of sound object. The sound object is defined as the basic unit of musical structure, which resembles the concept of note in traditional Western music. In other words, the basic unit of composition and analysis is the concept of sound object that encompasses sound events that are perceived as an entity. A sound object can be identified by its particular and intrinsic perceptual qualities that unify it as a sound event on its own and distinguishes it from all other sound events (Chion, 1983).

After enlightening seminal concepts and the methodology present in TOM, I will delve into the core of Schaeffer's treatise and present first a brief overview of the basic organization of PROGREMU, and then focus on the second stage of this program—morphology—which aims at outlining perceptual criteria for describing sound.

2.2.1 - Program of Music Research

The core of Schaeffer's TOM is the PROGREMU, which guides the user through "the art of practicing better listening" (Chion, 1983, p. 38) in relation to musical activity. PROGREMU is divided into five stages: (1) typology, (2) morphology, (3) characterology, (4) analysis, and (5) synthesis.

The first two stages of the PROGREMU are commonly addressed together as "typo-morphology," and as the most detailed stages undertake three tasks in relation to sound objects: (1) to identify; (2) to classify; and (3) to describe. The first task aims at identifying sound objects from an audio stream. The resulting segments are further classified into distinctive types, and, finally, exhaustively detailed according to their morphological characteristics. Typology takes care of the first two operations and morphology the third. In sum, sound objects are categorized into a typology based on perceptual attributes. While the ultimate aim of Schaeffer's typology is to assign "value" to sound object and derive their suitability for musical activity, morphology offers a refined and precise description of the sound objects and their inner structure. As

Schaeffer notes, typo-morphology is a “descriptive inventory which precedes musical activity” (Schaeffer, 1966, as cited in Chion, 1983, p. 124).

The next three stages of PROGREMU—characterology, analysis, and synthesis—are intrinsically related. Characterology’s purpose is to formulate “genres,” or, in other words, to define compound classes that articulate the morphologic criteria of sound objects into representative groups (Landy, 2007). Characterology introduces implicit notions related to instrumental terminology and groups sound objects that share the same musical value or specific features together. Analysis can be seen as a complementary stage of characterology in the sense that it examines the potential of sound objects to be arranged in “scales”, also referred as “species”, according to perceptual features (Landy, 2007). Finally, synthesis fills possible gaps in the available material for the composer by producing a “series of objects of the same genre leading to the emergence of a variation of a relevant feature, or value” (Chion, 1983, p. 114). Ultimately, synthesis provides to the sound based composer a larger pallet of raw material to assist the compositional process.

2.2.2 - Morphological Criteria of Musical Perception

Schaeffer’s morphological criteria for sound objects’ description are examined by human judgment through reduced listening. The morphological criteria are defined as “observable characteristics in the sound object” (Chion, 1983, p. 158), and “distinctive features […or] properties of the perceived sound object” (Schaeffer, 1966, p. 501), like the mass of a sound (e.g. sinusoidal or white noise), sound’s granularity and dynamics.

Two concepts—matter and form—organize Schaeffer’s morphology. For Schaeffer, if matter refers to the characterization of stationary spectral distributions of sound, then sound matter is what we would hear if we could freeze the sound. Form exposes the temporal evolution of the matter. Schaeffer studied matter and form by listening tests

that focused on sounds objects with two different natures: the first group encompassed sounds with fixed form to study the matter criterion, and the vice versa strategy on the second group of sounds, that is, a fixed matter that allowed him to study the form. In addition, Schaeffer provides a third category, called variation criteria, which analyzes the morphology of sounds in which both the form and the matter vary. These categories are further divided into seven criteria. A detailed description of each criterion follows.

Sound matter is characterized by the mass and harmonic timbre. Mass is the "mode of occupation of the pitch-field by the sound" (Schaeffer, 1966, as cited in Chion, 1983, p. 159). By examining the spectral distribution of a sound object, it is possible to define its mass according to classes that range from noise to a pure sinusoidal sound. Schaeffer divided the morphology of the mass criterion in seven classes: (1) pure sound: a tonic without harmonic timbre; (2) tonic or node: a mass characterized by a locatable pitch (e.g. a sinusoidal sound); (3) tonic group: a mass consisting of two or more tonics or nodes (e.g. a violin tone); (4) channeled sound: an ambiguous mass, composed of tonic, tonic groups, nodes, and nodal groups (e.g. piano chord); (5) nodal group: a mass formed of "bands" of mass (e.g. bell sounds); (6) nodal sound: a mass formed of an array of sounds which is non-locatable in pitch (e.g. sea sounds); (7) white noise: a complex mass that occupies the entire pitch-field (e.g. electronically-generated white noise).

Harmonic timbre, is closely related to the criterion of mass and complements it by further describing additional qualities of the mass (Schaeffer, 1966). The classes that characterize harmonic timbre are interdependent features of the mass' classes, and in certain cases it is even impossible to dissociate them (Chion, 1983). Schaeffer's classes of the harmonic timbre criterion are an extension of the classes of mass, in the sense they provide more details concerning the spectral distribution of each sound object. Instead of giving a deeper perspective of the classes presented earlier in the mass criterion, I prefer to inspect the characterology and analysis of harmonic timbre since it enlightens the nature of this criterion by using binary pairs of concepts regularly used in the musical

practice. Some of these binary concepts are: empty/full, round/pointed, resonant/dull, dark/light, narrow/broad, poor/rich (mass), and the density or volume.

Grain defines the microstructure of the sound matter, such as the rubbing of a bow. Even though it describes a temporal dimension of the sound, it is under the criterion of matter. Grain is divided into the following three classes: (1) resonance grain: non-sustained sounds (e.g. cymbal resonance); (2) rubbing grain: sustained sounds (e.g. bow or breath sounds); and (3) iteration grain: iterative sounds (e.g. drum roll).

Sound shape/form encompasses two criteria: (1) dynamic, and (2) pace. The dynamic criterion exposes and characterizes the shape of the amplitude envelope. Schaeffer distinguished several types of dynamic profiles (e.g. unvarying, impulsive, etc.), as well as several types of attack (smooth, steep, etc.). The pace (*allure*) defines the fluctuation in the sustainment of the spectrum of sound objects. It is analogous to the definition of the quality of the vibrato in instrumental or vocal sounds, or even, in electronic music, the attributes of amplitude or frequency modulations. Schaeffer (1966) defines three types of pace: (1) mechanical (very regular); (2) lively ("flexible periodicity, revealing a living being"); and (3) natural (unpredictable).

The variation criteria expose the temporal dimension of pitch and mass. The melodic profile is related to the variation of the pitch and is characterized by three variation types (imperfect stability, continuous—e.g. a glissando, and discontinuous—e.g. a piano phrase) and to three variation speeds (slow, medium, and fast). The mass profile denotes the continuous variation of mass, and is defined by several typical mass variations, such as pitch to complex or thin to thick.

2.3 – After Schaeffer

For more than four decades, TOM by Pierre Schaeffer had no deep implications and concrete consequences in musical analysis and particularly in the analysis of electronic

music (Thoresen, 2007b; Landy, 2007). Two of the most pointed reasons for this neglect are the difficulty of Schaeffer's writing and the unavailability of the document in English until 2004. According to Landy (2007), the first and only publicly available English translation is Schaeffer's short *Acousmatics* chapter published in Cox and Warner (2004). In this regard, a note should be paid to the impressive work of Michel Chion, whose *Guide des Objects Sonores: Pierre Schaeffer et la Recherche Musical* (1983) introduces Schaeffer's TOM and *Solfège des Objects Sonores* in a systematic fashion. Along with the work of Schaeffer there are two other sound-based theories—Denis Smalley's "spectromorphology" and Lasse Thoresen's aural sonology—that are seminal for this dissertation. Both theories acknowledge and extend the sound categorization provided by Schaeffer, in particular his morphological criteria of sound perception.

2.3.1 - Denis Smalley's Spectromorphology

Denis Smalley's spectromorphology is the most significant contribution to music theory related to sound categorization that acknowledges Schaeffer's influence (Landy 2007). Today, Smalley's spectromorphological criteria, which tremendously simplify and systematize Schaeffer's typo-morphology, are probably the most known and frequently applied theory for the analysis of electroacoustic music. Smalley presented his spectromorphological theory in two major articles: "Spectromorphology and Structuring Processes" (1986) and "Spectromorphology: Explaining Sound-Shapes" (1997). Smalley (1986) defines spectromorphology as "an approach to sound materials and music structures which concentrate on the spectrum of available pitches and their shaping in time" (p. 61). The term spectromorphology is a word formed through juxtaposition, referring to the interaction between sound spectrum (spectro-) and the way it evolves through time (-morphology) (Smalley 1997). Similar to Schaeffer's, Smalley's theory is not a composition treatise, but rather an analytical theory that covers a wide range of

concepts and terminologies for describing and studying the listening experience. Spectromorphology broadens the discussion regarding electroacoustic music analysis by proposing a systematic and shared terminology (1986). Smalley commonly refers to Schaeffer's typo-morphology, and even adopts reduced listening as his main research strategy.

Smalley's spectromorphology is divided into five basic classes: (1) spectral typology, (2) morphology, (3) motion, (4) structuring principles, and (5) space. He starts by offering an examination of the various sound types (spectral typology), and then moves into a broader description of the temporal shapes or morphologies of sound. Motion considers the organization of spectral shapes and the temporal discourse of the spectral-morphological design. Structuring principles, the fourth class, details how the listener experiences motion. Finally, Smalley's examination of space addresses the reception and interpretation of electroacoustic music by the listener; namely how the aural spaces that can be simulated with the use of spatialization technology constraint and create musical structures. For the purposes of this dissertation, the first two classes of spectromorphology—spectral typology and morphology—are the most relevant.

Spectral typology characterizes the spectrum of a sound according to a continuum whose boundaries are note (pure tone) and noise. Smalley (1986) points out that the concept of spectrum in electroacoustic music encompasses both the pitch and the timbre qualities of a sound (later, I address these separately in relation to instrumental and vocal music practices). The continuum of possibilities that this criterion defines is related to the density of the spectrum. The passage between notes to noise is a result of increased spectral density and compression. Smalley names this interval as "pitch-effluvium continuum." Smalley also segments the continuum into several typological components. The three main typological categories are: (1) note: a discrete pitch, which is further subdivided into: note proper, harmonic spectrum, and inharmonic spectrum; (2) node: a band of sound without clear pitch identification; and (3) noise: a spectrum with a highly

compressed density that inhibits the perception of any internal pitch structure.

Morphology denotes the temporal shaping of the spectrum. Smalley (1986) identifies and distinguishes three temporal phases of the sound objects: (1) onset, (2) continuant, and (3) termination. The articulations between these three phases produce a limited number of morphological models that expose a particular shape of the spectrum over time. Smalley provides a symbolic notation for each of the profiles, which depicts the temporal evolution of the shapes. Figure 2.1 offers a complete list of the profiles and their respective labels.

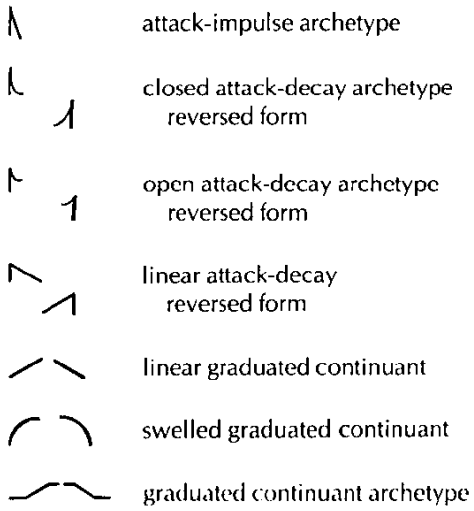


Figure 2.1 - Smalley's (1986) morphological profiles of sound objects. (Copyright 1986 by Palgrave Macmillan. Reproduced with permission.)

Smalley (1986) states that these morphological profiles can be linked or joined in strings in order to generate a wide and subtle variety of hybrid temporal articulations. Figure 2.2 illustrates these connections.

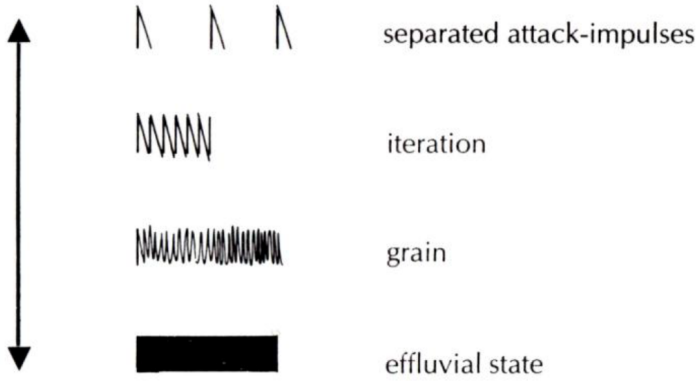


Figure 2.2 – Smalley's (1986) attack-effluvium continuum. (Copyright 1986 by Palgrave Macmillan. Reproduced with permission.)

Similar to the spectral typology's pitch-effluvium continuum, in morphology Smalley also presents an equivalent concept named "attack-effluvium continuum" that describes the range of possibilities offered by the rate and compression of the iteration between attack-impulses. Figure 2.2 depicts all possibilities within the attack-effluvium continuum. While the first two categories describe an iterative behavior with different time scales, the last two categories can be seen as a description of the sound's granularity—of which the ultimate stage (effluvial) is perceivable as a sustained sound.

2.3.2 - Lasse Thoresen and the Aural Sonology Project

The Aural Sonology Project (ASP) is a research program initiated in the 1970s at the Norwegian Academy of Music in Oslo by Lasse Thoresen with the assistance of Andreas Hedman and Olav Anton Thommessen. A major contribution of this ongoing project is an analytical framework for music for which no score is available, or music in which no simple one-to-one correspondence between score and the aural phenomenon exists (Thoresen, 2007a).

The project claims two main influences: the first is Sonology as taught at the Utrecht Institute of Sonology in The Netherlands, and the second, the typo-morphological point of view articulated by Pierre Schaeffer in TOM. ASP draws its fundamental principles on the primacy of the listening experience through reduced listening, a concept borrowed from Schaeffer (1966). However, ASP extends Schaeffer's theory towards "a pragmatic use of selected structuralist techniques" (Thoresen, 2007a). The musical object is apprehended not only as an objective fact but is partly formed by the listener's intentions.

The ASP developed an analytical approach to sound-based compositions with the following three levels: (1) sound objects, (2) elementary patterns, and (3) patterns of patterns. The first two levels—sound objects and elementary patterns—encompass constituent units or simultaneous layers of the sound continuum, and the third level characterizes the functional relationships between the several units (Thoresen 2007b). The following parallels can be established between Thoresen's concepts and the time scales defined by Roads (2001) and defined earlier: (1) sound objects is a common concept by both taxonomies; (2) elementary patterns and the meso time scale; and (3) patterns of patterns with the macro time scale.

One of the most valuable contributions of the project for the scope of this dissertation is the redefinition of Schaeffer's typo-morphology into a terminology suitable for describing the musical phenomenon in empirical terms and adapted to practical analysis. It relies on philosophical jargon, everyday language, and terminology from musicology and acoustics, employing terms that are not coined within a consistent phenomenological point of view. Bellow, I present the morphology proposed by Thoresen that is divided in four main criteria, each of which further subdivide into smaller classes: (1) sound spectrum, (2) dynamic profile, (3) gait, and (4) granularity.

Sound spectrum is characterized by spectral width and spectral brightness. The width of the spectrum characterizes the magnitude of the spectral components of a signal according to a continuum of possibilities, whose boundaries range from pure tones to

white noise. Within these limits an endless set of possibilities can be described such as monophonic and polyphonic pitched sounds with harmonic or inharmonic spectrum, or even any further saturations of the sound spectrum from unrecognizably pitched structures to a fully saturated spectrum (white noise). Spectral brightness indicates the spectrum's "center of mass" and has a strong connotation with the perception of the "color" of the sound; it is commonly described by adjectives such as "dark" or "bright." As Thoresen (2007b) notes, spectral brightness is a well-known phenomenon in linguistics to discern and organize the vowels and consonants, and in the music domain is a crucial feature for distinguishing between several traditional musical instruments, for instance.

Dynamic profile "expresses the energy articulation of a sound object" (Thoresen 2007b). Thoresen proposes the following seven profiles, based on Schaeffer's typology, to characterize the dynamic shape of sound objects: (1) no dynamic profile (*dynamique nulle*); (2) weak dynamic profile (*dynamique faible*); (3) formed dynamic profile (*dynamique formée*); (4) impulse-like dynamic profile (*dynamique-impulsion*); (5) cyclic dynamic profile (*cyclique*); (6) vacillating dynamic profile (*rëiteré*); (7) accumulation-like dynamic profile (*accumulé*). He (2007b) further characterizes the dynamic profiles by providing a typological description of two distinct phases of each profile: the onset and the termination. There are seven onset types and six termination types, which will not be detailed here, because the extrapolation of a sound typology is not relevant for the scope of this dissertation. The onset phase can be even further characterized by adding an indication of the spectral brightness of the opening transient.[7]

Gait is Thoresen's attempted translation of the French word *allure*, one of the criteria of Schaeffer's typo-morphology. Gait is closely related to the idea of vibrato, and defined by Thoresen (2007b) as "the undulating movement or characteristic fluctuation that often can be found in the sustained part of sound objects" (p. 139). Thoresen further divides the gait criterion in three categories according to nature of the undulation, which can be

[7] For a detailed description of typology of the onsets and terminations please refer to Thoresen (2007b).

traced in the pitch, dynamic, or spectrum dimensions. Each category can be addressed by their nature, namely, pitch gait, dynamic gait, and spectral gait. Gait can be also characterized according to the degree of the undulation's deviation from its mean value (small, moderate, and large) and the pulse velocity of the undulation (slow, middle, and fast).

Granularity describes the microstructure of a sound object, that is, the perceptual irregularities. It is analogous to the abrasiveness one can feel when touching a piece of cloth or material, or the visible granularity of a photograph. Granularity is intrinsically related to the notion of iteration, and it is hard to differentiate the two concepts. "Generally, grains are a micro feature of the object in question, whereas iterations are of a coarser kind; thus grains would tend to be smaller, quicker, and be inseparable from the main body of the sound" (Thoresen, 2007b). Thoresen distinguishes nine types of sound objects' granularity, which result from the combination of two characteristics of the grains: coarseness (small, moderate, and large) and velocity (slow, middle, and fast).

2.4 – Critical Review and Summary

This chapter reviewed three major analytical theories by Pierre Schaeffer, Denis Smalley and Lasse Thoresen for sound-based compositions. Special attention is given to the typological and morphological criteria to describe sound objects because they support a major contribution of this dissertation, which is a computational description scheme for sound objects. Smalley's and Thoresen's theories are rooted in the seminal work of Pierre Schaeffer, whose typo-morphology has considerably inspired several approaches in music analysis and composition—like Smalley's spectralmorphology and Thoresen's ASP.

The three aforementioned theories—typo-morphology, spectralmorphology, and aural sonology—were developed for the analysis of electroacoustic music. However, as Smalley (1986) points out in regards to his own theories, they are easily extensible to other music

genres. A GMR researcher named François Delalande asserts a position that may help us further understand the application of such theories to the analysis of music. While referring to Schaeffer's solfeggio, Delalande (1998) notes that:

> The morphological analysis of electroacoustic music (based on a resolution into sound objects) is a 'syllabic' analysis, which does not provide the means of highlighting pertinent configurations either poietically (a 'trace' of compositional strategies) or aesthesically (contributing to explaining the behaviours and representations of listeners). Thus, we do not consider a morphological analysis to be a music analysis. (p. 20)

As descriptive properties, the information provided by the analytical theories should not be treated as ends unto themselves, but rather as intermediary characteristics of sound objects. Moreover, the theories, and particularly the criteria for sound description, are broad enough to not be restricted to any music genre or style.

In music practice, the application of Schaeffer's typo-morphology has been nearly inexistent. Among the existent theories, Denis Smalley's spectromorphology has received the most attention from the music community. Lasse Thoresen has taken the inaccessibility of Schaeffer's work and presented a simpler, yet systematic, model that synthesizes Schaeffer's major classes and enhanced applicability, providing a symbolic notation for each descriptor.

The following paragraphs provide a critical review of the description schemes of the three aforementioned authors and establish a comparison between criteria used by them. A comparison between the top-level criteria of the three sound-based theories is provided in Table 2.1. In the following paragraphs, I adopt Schaeffer's typo-morphology as the basis for the discussion.

	SOUND MATTER			SOUND SHAPE/ FORM		VARIATION CRITERIA	
Pierre Schaeffer	Mass	Harmonic-timbre	Grain	Pace (*alure*)	Dynamic criteria	Melodic profile	Mass profile
Denis Smalley	Spectral typology		Morphology (*attack-*) *effluvium continuum*		Morphology		
Lasse Thoresen	Sound spectrum		Granularity	Gait	Dynamic profile		
	Spectral width	*Spectral brightness*					

Table 2.1 - Comparison between criteria of music perception of three representative sound-based theories by Pierre Schaeffer, Denis Smalley, and Lasse Thoresen.

Schaeffer's criterion of mass is present in both description schemes by Smalley and Thoresen, under the designations spectral typology and sound spectrum (more precisely spectral width), respectively. The three criteria are very similar, in particular the categorization of sound according to discrete types, whose limits are pure tones and full-saturated spectra. Alongside the description of the sound objects' mass by types of sounds, Smalley also adopts a description of the sound objects' mass in a continuum of possibilities, whose limits are note and noise.

Harmonic timbre is probably the most ambiguous criterion presented in Schaeffer's morphology. Its definition is very vague and its close relation with the mass criterion is somehow misleading. Smalley avoids this criterion altogether and Thoresen presents a sound descriptor that clearly belongs to the harmonic timbre criterion within the mass criteria (sound spectrum according to Thoresen's terminology).

All theories examine the grain or granularity of sound objects. However, while Schaeffer and Thoresen consider it as criterion on its own, Smalley describes this dimension within the morphology criterion, particularly in the attack-effluvium

continuum. Half of the interval of the attack-effluvium continuum, between grain and effluvial states, can be seen as a description of the granularity of the sound.

Another ambiguous concept presented in Schaeffer's TOM is the notion of pace (*allure*). Similarly to harmonic timbre, Smalley avoids this criterion. Thoresen adopts the criterion and enlightens its definition by providing simpler, yet reliable categories for describing both the nature (pitch, dynamic, and spectral), and the quality of the phenomenon (velocity and amplitude of the undulation). Still, I find Thoresen's definition of pace unsystematic and inconsistent, namely having in mind its algorithmic implementation, since it does not offer a concise description of the limits of the criteria.

The dynamic criterion is transversal to the three frameworks, even if some nuances may distinguish them. All authors give priority to the description of the amplitude attack. Schaeffer and Smalley focus on the description of the overall stability of the sound objects' dynamic, as well as the type of attack. Thoresen further explores some harmonic timbre characteristics of the attack phase.

A final comment should be addressed to the simplifying approaches of the frameworks that follow and acknowledge Schaeffer's theory. Even though it is seminal to understand the roots of Smalley's and Thoresen's theories (i.e., Pierre Schaeffer's TOM), their contributions provide a much better adapted framework for the ultimate goal of this discussion, which is the formulation of a computational scheme for the description of sound objects.

Chapter 3

Computational Segmentation and Description of Sound Objects

The current chapter aims at presenting strategies for segmenting and describing sound objects by computational means. It starts by providing an overview of techniques and tools used in MIR for the computational description of audio signals (§ 3.2), and ends by proposing algorithmic strategies for segmenting an audio stream into sound objects (§ 3.3) along with a musician-friendly description scheme that intends to characterize sound objects according to perceptual criteria (§ 3.4). The description scheme is particularly adapted to musical imperatives and targets a musicians' audience by relying on the interaction between MIR and musicological literature.

3.1 – Introduction

The ever-increasing amount of digital audio made available through public and private databases has demanded a deeper understanding of audio signals, in particular the

formulation of algorithms that can automatically extract information from audio data. Content-based audio processing is a recent technology designed to address the problem of sound indexing—offering new functionality for browsing, interacting, rendering, personalizing and editing musical material—by automating the task of manually annotating large sound databases. Most content-based audio processing research focuses on the recognition of sound sources (Martin, 1999; Eronen, 2001; Herrera, Dehamel, & Gouyon, 2003; Wold, Blum, Keislar, & Wheatom, 1996; Misdariis, Smith, Pressnitzer, Susini, & McAdams, 1998), music classification (Lu, Jiang, & Zhang, 2001; Tzanetakis & Cook, 2001; Ellis, Whitman, Berenzweig, & Lawrence, 2002), and music recommendation (Cano et al., 2005).

Research in, and attention to, this field increased significantly when the Moving Picture Experts Group (MPEG), started working around 1996 on MPEG-7, a standard for describing multimedia content. Unlike their previous standards, which were mostly codecs for multimedia content, MPEG-7 targeted the creation of standardized descriptions for multimedia data, along with ways for structuring them (Herrera, Serra, & Peeters, 1999; Kim, Moreau, & Sikora, 2005). The primary purpose of MPEG-7 is to easily allow users or agents to search, identify, filter, and browse audiovisual content. MEPG-7 represents audio signals using audio descriptors—a research topic that has captured a lot of attention from the MIR community and consequently extended how computers manage audio.

3.2 - Audio Descriptors

A critical feature of systems that deal with content-based audio processing, at any level, is the selection of audio data representations. The output quality of these systems is commonly dependent on such representations. The most common approach to represent audio in such systems is the adoption of audio descriptors, which measure properties of audio signal content and wrap audio features to sets of values. For example, the

brightness of a sound can be extracted by the audio descriptor spectral centroid, which measures the center of mass of the time-domain representation of an audio signal and expresses the brightness of a sound in a single value. Despite the numerous developments in this area, even state-of-the-art technology cannot compare with the accuracy, fastness, and detail of human perception and cognition.

The computation of audio descriptors involves the use of various and sometimes overlapping approaches. Not even in the context of the MPEG-7 is there a standard way of obtaining these descriptions, or a customary approach on how to use them (Herrera et. al, 1999). Some of the most common techniques for extracting relevant features from audio data are through signal processing, computational auditory scene analysis, and statistics. (Herrera et. al, 1999). Despite the idiosyncrasies of the various audio description approaches, there are common taxonomies applied.

Descriptors can be classified according to the representation of their output as well as their level of abstraction. According to Schwarz (2000), audio descriptors can be organized into three different classes: (1) categorical (class membership); (2) static (a single value); and (3) dynamic (temporal evolution). Specific to the level of abstraction, audio descriptors can also be organized according to the following three categories: (1) low-level, (2) mid-level, and (3) high-level.

Low-level descriptors are computed from the digitized audio data by simple means and with very little computational effort in a straight or derivative fashion. Literature in signal processing and speech processing documents an enormous amount of different low-level features that can be computed from the audio signal representation, either on the time domain (e.g. amplitude, zero-crossing rate, and autocorrelation coefficients), or on the frequency domain (e.g. spectral centroid, spectral skewness, and spectral flatness) (Schwarz, 2000). Most low-level descriptors make little sense to humans, especially if one does not master statistical analysis and signal processing techniques, because the

terminology used to designate them denotes the mathematical operations on the signal representation.

Mid-level descriptors require some level of interpretation from the data. This group of descriptors infers information directly from the audio data or from the results of a prior analysis, and usually relies on machine learning algorithms and statistical analysis. Known descriptions that fall within this group are chord, key, and meter induction. One of the downsides of this descriptor group is the time-consuming learning phase that most of the algorithms require (Gouyon et al., 2008).

The jump from low- or mid-level descriptors to high-level descriptors requires bridging a semantic gap. High-level descriptors, also referred to as user-centered descriptors, express some categorical, subjective, or semantic attributes of the sound, such as mood and genre. The computation of high-level descriptors involves some level of learning that has to be carried by means of a user-model and not only a data-model (as is the case of mid-level descriptors). As an example, let us imagine a simplistic "mood" descriptor consisting of labels "happy" and "sad." In order to automatically annotate unknown audio sources with such labels, it is necessary to initially create a computational model that distinguishes the two classes—happy and sad—by commonly relying on human annotations of audio tracks and machine learning algorithms; and then, one would be ready to compare a new track against the created model to infer its mood (Gouyon, et al., 2008).

3.2.1 - Audio Description Schemes Inspired in Schaeffer's Typo-Morphology

There is a line of research within the content-based audio processing field that is particularly relevant for the scope of this study because it relies on the same basis as the description scheme I intend to propose; that is, the musicological work of Pierre Schaeffer. Three computational description schemes follow this approach. I am referring to the work of Julian Ricard (2004), Geofroy Peeters and Emmanuel Deruty (2008), and

Norbert Schnell et al. (2010), respectively. While the first two offer a general framework for sound description, without considering specific applications, the latter framework was devised with the purpose of assisting the creation of the Marco Antonio Suárez Cifuentes' composition *Caméléon Kaléidoscope* (2010) for an ensemble of 15 instruments and electronics.

All three authors claim that their primary motivation for departing from Schaeffer's theory, which I also share, is the ability of his criteria to describe any type of sound independent of sources or causes. In addition, Schaeffer's criteria limits the number of perceptual characteristics for sound description to a reduced number of descriptors, an attitude opposed to traditional methodological approaches in MIR that use the largest possible number of descriptors in order to achieve better results. Nonetheless, Schaeffer's criteria encompass the most relevant parameters for sound description worth exploring in music analysis and composition.

Table 3.1 not only provides the complete set of descriptors adopted in each of the description schemes of the three aforementioned authors, but also provides a comparison between them. The (horizontal) top layer organizes the descriptors according to the three major perceptual criteria in Schaeffer's theory, as a way of establishing a bridge to their unifying root. Note that none of the three sets of criteria presents an exhaustive computational framework of Schaeffer's PROGREMU. Instead, their focus is mainly restricted to the first two stages of the PROGREMU—the typo-morphology— that moreover are appropriated with relative freedom, due (from my perspective) to the difficulty to understand some criteria presented by Schaeffer and/or their subjectivity.

	SOUND MATTER			SOUND SHAPE/ FORM	VARIATION CRITERIA			
Schnell et al.	Most salient pitch/ Pitchness	Centroid / Standard deviation		Loudness	Loudness Envelope			
Peeters and Deruty	Pitch	Spectral distribution	Texture		Dynamic profile		Duration	Space
Ricard	Pitchness	Brightness	Roughness		Dynamic profile	Pitchness profile	Duration	

Table 3.1 - Comparison between computational schemes for the description of perceptual attributes of sound inspired by Schaeffer's typo-morphology.

Of notice in all schemes present in Table 3.1 is the lack of harmonic timbre descriptors, which, given its the importance in recent composition practices, seems a significant absence that I will attempt to solve in my description scheme. But before addressing my description scheme, I must detail the strategies I have employed to computationally identify and segment an audio stream into sound objects.

3.3 – Identifying Sound Objects Computationally

The segmentation of an audio stream into sound objects cannot rely on the premise that was described in the beginning of this chapter—that is, the formulation of computational strategies based on musicological literature—because a systematic musicological approach for the segmentation of a sound continuum into sound objects does not exist (Smalley, 1986). Therefore, the groundwork that supports the segmentation strategies will solely rely on MIR literature. In particular, algorithms for onset detection

and audio beat tracking. A description of the solutions adopted in my framework and implemented in my software earGram follows.

3.3.1 - Onset Detection

Onset detection algorithms find the location of notes or similar sound events in the audio continuum by inspecting the audio signal for sudden changes in energy, spectral energy distribution, pitch, etc. A large amount of MIR literature has been devoted to onset detection algorithms over the last few decades[8] because it is widely used as a pre-processing stage in many applications such as: automatic transcription (Bello & Sandler, 2003), annotation (Tzanetakis & Cook, 1999), sound synthesis (Schwarz, 2004; Jehan, 2005), or rhythm and beat analysis (Uhle & Herre, 2003; Goto, 2001).

Onset detection is commonly computed in two steps. The first step is a pre-tracking stage and encompasses the computation of a continuous periodicity function, which expresses audio features (e.g. energy, phase, or pitch) along time lags. The second step attempts to find discontinuities or abrupt changes in the descriptor(s) function(s). Current algorithms for onset detection adopt quite distinct audio features, or combinations of them, in order to convey better results for specific types of sounds, such as percussive, pitched instrumental, or soundscapes (Bello, Duxbury, Davis, & Sandler, 2004; Dixon, 2006). Therefore, in order to address a variety of sounds, I adopted and implemented three distinct onset detection algorithms in earGram. As Bello et al. (2004) state while referring to the choice of an appropriate onset detection method, "the general rule of thumb is that one should choose the method with minimal complexity that satisfies the requirements of the application" (p. 1045). Two of the algorithms (named "onset1" and "onset2") inspect the audio for abrupt changes in the spectral energy distribution, and the third ("pitch") detects different fundamental frequencies. The first two onset detection

[8] Please refer to Bello, Duxbury, Davis, and Sandler (2004) and Paul Brossier (2006) for an exhaustive review of onset detection algorithms.

functions cover the totality of perceivable sounds, but despite using a common audio feature in the pre-tracking stage, they present levels of sensitivity very distinct and therefore target sounds with different natures. Onset1 provides better results for environmental sounds and onset2 for musical sounds. Pitch is limited to the segmentation of pitched monophonic sounds, and its implementation is due to the significant improvement in the detection of onsets in this type of audio signals.

Onset1 and onset2 use the spectral flux as the audio feature over which all further processing is done. The choice of this descriptor relies on recent comparative studies evaluating alternative onset-detection functions (Dixon, 2006). A slight difference between the two algorithms relies on the spectral representation used: onset1 computes the spectral flux on the magnitude spectrum, and onset2 wraps the spectrum representation into a perceptually determined Bark frequency scale, which resembles the spectral information processing of human hearing.

I utilized two onset detection strategies based on the same audio feature, mainly because their peak-peaking stage is considerably different. The peak detection phase of onset1 is rather simple and reports onsets when it detects local maximum values above a threshold value, after falling below a low threshold. The implementation of onset1 relies on code by William Brent (2011). The peak detection stage of onset2 algorithm reports onsets in a similar fashion as onset1, that is, by selecting local maxima above a higher threshold value; however, it adopts some processing on the descriptor function, and the threshold is assigned in a dynamic manner. The adoption of this peak-peaking pre-processing stage was proposed by Paul Brossier (2006) to limit the number of spurious peaks in the detection function. To reject false positive detections in areas of low energy, onset1 and onset2 segmentation strategies adopt a simple envelope detector at the end of the processing chain that discards onsets below a given loudness threshold.

Pitch defines units by slicing the audio continuum at the beginning of notes. The processing relies essentially on a pitch detection algorithm developed by Puckette, Apel,

and Zicarelli (1998). Some post-processing is applied to the algorithm to ignore sudden jumps in the analysis if their adjacent analysis windows report a stable pitch. In addition, detected notes need to be at least two analysis windows apart.

3.3.2 - Audio Beat Tracking

The aim of an audio beat tracking algorithm is to find an underlying tempo and detect the locations of beats in audio files. The task corresponds to the human action of tapping a foot on perceptual music cues that reflect a locally constant inter-beat interval. The topic is extensively discussed in MIR literature, and various algorithms that achieve quite remarkable results have been presented in the last decades (Davis & Plumbley, 2007; Ellis, 2007; Oliveira, Gouyon, Martins, & Reis, 2010). Audio beat tracking is often mentioned as one of the solved problems in MIR, however there are still unresolved issues—namely handling complex times, extremely syncopated music, and long periods of silence.

While providing a review of existing algorithms for audio beat tracking is out of the scope of this dissertation,[9] it is important to understand the general building blocks commonly adopted in audio beat tracking algorithms, which served also as a basis for the my audio beat tracking algorithm implemented in earGram: (1) audio feature extraction, (2) beat or pulse induction, and (3) beat tracking per se.

The starting point of most computational models for beat tracking is the extraction of features from the audio signal that carry relevant rhythmic information (e.g. amplitudes, pitches, and spectral flux). The second step infers the beat by finding periodic recurrences of features in time. And finally, the output of the second step feeds the third processing stage, which attempts to find the beat in the audio data. Although most algorithms assume that the pulse period is stable over an entire song, many algorithms take into account timing deviations, which commonly result from errors or expressivity. Gouyon and

[9] Interested readers are referred to Gouyon and Dixon (2005) for a comprehensive review of rhythm description systems and in particular beat tracking algorithms.

Dixon (2005) point that the computation of short-term timing deviations is particularly relevant when attempting to find the location of beats.

I had to implement a new algorithm for offline audio beat tracking in earGram because there are no available tools in Pure Data (earGram's programming environment) to compute such task. Initially, my algorithm infers the tempo (beats per minute) of audio data stored in a buffer by finding the highest value of the accumulated spectral flux autocorrelation function. Then, in order to find the beat location, my algorithm starts by selecting the ten highest peak values of the spectral flux function (i.e., the ten onsets with higher growth values), and, relying on my hypothesis that one of these ten onsets corresponds to a beat location, the algorithm inspects for each selected onset the location of the beats according to the induced tempo. The computation of the beat locations allows short-term timing deviations, only if a local maximum is found within 2048 tolerance samples from the predicted location. For each of the ten onsets a score is computed by accumulating the spectral flux values from each prediction. Finally, the beat locations with the highest score are reported.

After the segmentation of user-assigned audio tracks into sound objects, earGram extracts meaningful information from the sound objects' audio signal representations and provides feature vectors that exposes their most prominent characteristics. The audio descriptors used to extract features of the audio will be detailed in the remaining sections of this chapter.

3.4 - A Musician-Friendly Audio Description Scheme

In the creation of the description scheme that I will detail in this section, I relied on eight premises (formulated before its creation) to guide, unify, and regulate the set of perceptual criteria devised. In order to clarify the guidelines that assisted the creation of the description scheme utilized in earGram, the following premises are presented to the

reader.

Some of the guiding principles of the description scheme were particularly devised to convey its primary use, the characterization of audio units of a CSS system (earGram). However, even if the scheme addresses idiosyncratic features of CSS, its application context is not restricted to this synthesis technique. The scheme encompasses dimensions that can be easily adapted to application contexts that require sound descriptions regardless of the relation between the sonic phenomenon and its source. Premises one to five address general considerations of the scheme, and premises six to eight address the idiosyncratic aspects of CSS.

1) The applied terminology in the scheme should rely on concepts from music theory and practice, in order to offer a more user-friendly experience for people with a music education background.

2) It should promote musical activity, specifically by providing representations of audio signals that can be easily manipulated in CAAC strategies.

3) It must rely solely on the abstract perceptual characteristics of sound—the morphology of sounds—disregarding their source, means of production, or stylistic features.

4) The descriptors' computation should be definable by a mathematical function.

5) It should consider the emergence of higher-level descriptions of audio signals by associating or manipulating the basic criteria proposed in the scheme.

6) It should cover a continuum of possibilities and avoid the lattice-based organization of sound units (Wishart, 1994). Every criterion should be defined in a linear continuum with limited typological categories of sounds. This feature is appropriated from Smalley's spectromorphology, namely its pitch- and attack-effluvium continuums.

7) All descriptors must have the same range.

8) The descriptions should be invariable to the units' duration. In other words, the descriptions should allow meaningful comparisons between units of different durations within the same time scale.

Relying on the eight premises listed above, I started to devise the top-level organization of the description scheme, which relies on two concepts borrowed from Schaeffer: matter and form. While the criteria related to matter describe the units' sound spectrum as a static phenomenon, the form criteria expose the temporal evolution of the matter.

The matter criteria express features of the audio in numeric values in a linear continuum interval, whose limits correspond to typological categories; the form criteria are expressed as vectors. In other words, the matter criteria represent each sound object with a numerical value, which is meaningful in relation to a finite space whose limits represent particular types of sounds. The form criteria follow the same approach but provide a contour of the audio features' evolution. For example, noisiness, a criterion of matter, describes sound objects in relation to two typological limits (pure tone and white noise), and within these limits, sound objects are defined by a numerical value according to its characteristics. Sound typologies (as defined by Schaeffer) are only used here to define the limits of the interval. The dynamic profile, in turn, exposes the evolution of the amplitude of a sound object.

Matter is further divided in two other categories: main and complementary. While the criteria under the main category provide meaningful descriptions for the totality of sounds that are audible to humans, the criteria under the complementary category provides meaningful results for limited types of sounds. For example, pitch—a complementary criterion of mass—only provides meaningful results for pitched sounds, thus excluding all sounds that do not fall in this category.

	MATTER		FORM
	MAIN	COMPLEMENTARY	
Mass	Noisiness	Pitch	
		Fundamental bass	
			Spectral variability
Harmonic Timbre	Brightness		
	Width		
	Sensory dissonance		
	Harmonic pitch class profile		
Dynamics	Loudness		Dynamic profile

Table 3.2 - Description scheme used to characterize the audio content of sound objects in earGram.

In choosing the descriptors that constitute the scheme, I relied on three musicological theories presented earlier—Schaeffer's typo-morphology, Smalley's spectromorphology, and Thoresen's aural sonology—but I did not fully incorporate them into the scheme because of simplicity, usability, and/or technical reasons. Instead, I selected the ones that are more adapted to the practice of music composition, and whose technical implementations are feasible. A major concern behind the description scheme was the use of terminology from music theory and practice. Therefore, without disregarding the use of concise concepts, the terms used in the description scheme attempt to facilitate the usability for musicians with a traditional Western music education.

While the conceptual basis of the scheme is entirely mine, the computation of each descriptor relies on algorithms from others—in particular William Brent's timbreID library (Brent, 2009)—to extract low-level audio features from the audio. I chose Brent's library for its robustness, efficiency, and ability to work in both real-time and non-real time. The detailed conceptual basis and technical implementation of each description in my scheme follows.

3.4.1 - Criteria of Mass

The mass criteria examine the spectral distribution of a sound object in order to characterize the organization of its components. It not only attempts to detect spectral patterns (e.g. pitch, fundamental bass) but also to provide general consideration of the spectral distribution (e.g. noisiness). The criteria of mass encompass four descriptors: (1) noisiness, (2) pitch, (3) fundamental bass, and (4) noisiness profile. The first is a main descriptor of matter, the second and third are ancillary descriptors of matter, and the last descriptor falls into the form category.

3.4.1.1 - Noisiness

The noisiness descriptor estimates the amount of noisy components in the signal as opposed to pitched components. The measure of noisiness is present in all theories described in Chapter 2. However, each theoretician adopts a different term for this descriptor: Pierre Schaeffer designates it as mass; Denis Smalley uses spectral typology; and Lasse Thoresen names it spectral width. I adopted the term noisiness in the detailed scheme, because not only is it an easily understandable concept by both experts and non-experts, but also because it is related to the algorithmic nature of the descriptor (detailed below). Another feature of the noisiness descriptor inspired by Smalley's musicological theory is the adoption of a linear continuum to characterize the sound objects.

The noisiness descriptor is calculated as a weighted sum of the following four low-level descriptors: (1) spectral flatness, (2) spectral kurtosis, (3) spectral irregularity, and (4) tonalness. Spectral flatness is the most significant descriptor, while spectral kurtosis, spectral irregularity, and tonalness are primarily useful to provide a better distinction between pitched sounds and noisy sounds.

Spectral flatness is a very robust indicator of the noisy components of a signal, and

provides reliable descriptions of all sounds. However, it poses a major and pertinent problem: its characterization of pitched sounds is extremely poor. In other words, spectral flatness has a very good resolution for noisy-like signals; however, it is quite crude in relation to pitched sounds. In some experiments I carried out with a corpus of heterogeneous sounds, I immediately noticed a discrepancy between the interval of pitched sounds, which fall roughly in the interval]0, 0.1[, and the interval that comprises noisy sounds, which inhabit the rest of the scale. As mentioned, in order to reduce this problem, I merged the results from spectral flatness with three other descriptors—(1) spectral kurtosis, (2) spectral irregularity, and (3) tonalness. A brief definition of each of the aforementioned descriptors, along with their contribution to enhance the representation of pitched sounds in the noisiness criterion follows.

Spectral kurtosis gives a measure of the flatness or "peakedness" of the spectral distribution around its mean value (Peeters, 2004). The kurtosis of a single sinusoid will be extremely high, while that of noise will be extremely low. Spectral kurtosis is particularly good at distinguishing between pitched sounds that range from pure tones to heavy frequency modulations.

Spectral irregularity enhances the difference between jagged and smooth spectra by looking at the spectrum from low to high frequencies and denoting how each bin compares to its immediate neighbors. Spectral irregularity has two common definitions: one by Jensen (1999) and other by Krimphoff, McAdams, and Wimsberg (1994). For practical reasons, I use Jensen's measure since it defines the irregularity of a spectrum by values between zero and one, avoiding further processing to convey the same numeric interval used in the other descriptors. For jagged spectra (e.g. tone with harmonic spectra), irregularity will be high, and for smooth contoured spectra (e.g. filtered noise) it will be low. Spectral irregularity enhances the distinction between sounds from tones with harmonic or inharmonic spectra to spectral distributions formed of several "bands" (e.g. bell sounds) to spectral distributions formed of an array of sounds, which is non-locatable

in pitch (e.g. sea sounds).

Tonalness measures the "perceptual clarity of the pitch or pitches evoked by a sonority" (Parncutt & Strasburger, 1994, p. 93). Sounds with high tonalness values evoke a clear perception of pitch. I use the tonalness measure by Parncutt, which defines the (pure) tonalness as the quadratic sum of the spectral pitch weights (so that its maximum value is equal to one). The tonalness descriptor I use is a slightly altered version of code provided in the Dissonance Model Toolbox by Alexandre Porres (2011).

The combination of descriptors detailed above enhances the quality of the noisiness criterion by providing a better definition of pitched sounds and a better distinction between pitched and noisy sounds. Still, I applied some additional processing to each descriptor individually to further enhance the balance between pitched and noisy sounds. Spectral flatness and spectral kurtosis were scaled by exponential and logarithmic functions, respectively. No post-processing was applied to the spectral irregularity and tonalness descriptors.

The noisiness descriptor ranges between zero and one. Zero represents a full saturated (noisy) spectrum and one represents a pure sinusoidal without partials. Within these two extremes the descriptor covers the totality of audible sounds including instrumental, vocal, or environmental sounds.

3.4.1.2 - Pitch

The name of the second descriptor of mass is self-explanatory; it reports the pitch or fundamental frequency of the units. Pitch is a secondary criterion of mass, since it only conveys meaningful results for pitched sounds, and thus may reduce the corpus to a smaller collection of units. This descriptor is not contemplated in any theory discussed in Chapter 2 because it is highly attached to the concept of musical note and does not provide meaningful descriptions for the totality of perceivable sounds. However, the pitch

descriptor is adopted here since it may constitute an extremely important element in the composition process when dealing with pitched audio signals.

There are several robust algorithms to estimate the pitch or fundamental frequency of monophonic audio signals. State-of-the-art algorithms for polyphonic pitch detection are not very reliable. The Pure Data's built-in object sigmund~ by Miller Puckette is the pitch detection algorithm used to compute the fundamental frequency of (monophonic) sounds. The output of the descriptor is twofold: (1) in MIDI note numbers and (2) pitch classes. I additionally scale the resulting values to the interval [0, 1] to convey the general range of all descriptors.

3.4.1.3 - Fundamental Bass

The fundamental bass descriptor reports the probable fundamental frequency or chord root of a sonority. Similar to the pitch criterion, it is a secondary criterion of mass, because it may reduce the corpus to a smaller number of units. I utilize this descriptor as a strategy to overcome the limitation of the pitch descriptor when analyzing polyphonic audio signals.

The fundamental bass is computed by an altered version of a Pure Data object from the Dissonance Model Toolbox by Alexandre Porres (2011). The fundamental bass corresponds to the highest value of the pitch salience profile of the spectrum. The pitch salience of a particular frequency is the probability of perceiving it or the clarity and strength of tone sensation (Porres, 2012). The fundamental bass is expressed in (1) MIDI note numbers and (2) pitch classes. The output of the descriptor is further scaled to the interval [0, 1].

3.4.1.4 – Spectral Variability

Spectral variability provides a measure of the amount of change in the spectrum of an audio signal. It is computed by the low-level audio descriptor spectral flux (Peeters, 2004), which compares adjacent frames by calculating the Euclidean distance between two non-normalized spectra. The use of non-normalized spectra not only accounts for spectral differences, but also denotes sudden amplitude changes. Spectral variability is a form descriptor since it describes a temporal phenomenon.

The output of this descriptor is threefold: (1) a curve denoting the spectral variability of the unit, (2) basic statistical values that express characteristics of the curve (such as maximum and minimum values, mean, standard deviation and variance), and finally (3) a single value that expresses the overall spectral variability throughout the unit duration. The curve depicts the evolution of the spectrum at regular intervals of 1024 samples, and each analysis window encompasses 2048 samples.

Relying on the computed curve some basic statistical properties are then extracted, such as minimum, maximum, mean, standard deviation, and variation. These statistical properties provide a characterization of the curve by a vector with a reduced dimensionality. A single value depicting the overall variability of the overall unit's spectrum is computed in four steps: (1) dividing the units in two equal halves, (2) computing the spectrum of each half, (3) calculating the distance between the two spectral representations, (4) summing all values resulting from point 3. The output of the descriptor is further divided by the number of reported bins (resulting from the spectral difference computed in point 3 to scale the output to the interval [0, 1].

3.4.2 - Criteria of Harmonic Timbre

The three musicological theories presented earlier provide little guidance for the formulation of algorithmic strategies to describe the harmonic timbre content of a signal. Schaeffer's criteria of harmonic timbre are very misleading and too inconsistent to be encoded algorithmically. Smalley (1986, 1997) does not provide a specific set of criteria for harmonic timbre; even if he considers this dimension while describing the mass of sound objects under spectral typology. Thoresen's sound spectrum criteria, in particular the spectral brightness, are the most adapted to a computational definition of harmonic timbre. His criteria also points towards the possibility of including psychoacoustic models as harmonic timbre descriptors, which Schaeffer rejected because (in his opinion) the in vitro psychoacoustic experiments do not fully apprehend the multidimensionality qualities of the timbre (Chion, 1983). Still, Thoresen's suggestion led me to further investigate psychoacoustic literature, most notably models that examine the sensory dissonance phenomenon, which provide a good description of spectra distributions.

The main source for investigating possible usages of psychoacoustic models in my description scheme was Alexandre Porres' PhD dissertation (Porres, 2012). Porres not only explores several creative applications of psychoacoustic models in signal processing, but also points out the underexplored possibilities of psychoacoustic dissonance models for the automatic description of audio signals' content. This has not been subject to any study and could contribute significantly to applications such as CSS and alike (Porres, 2012).

> While concatenative sound synthesis and similar techniques are common and are at an interesting development stage, processes of the same order with the higher-level descriptors, such as the attributes of dissonance here exposed, have not been fully explored, with the exception of some computer-assisted composition works by Sean [Ferguson] (2000) and [Clarence] Barlow (1980). (Porres, 2012, p. 86)

Therefore, I will use here the psychoacoustic dissonance models presented by Porres

(2012), which are largely based in Terhardt's psychoacoustic theory (Terhardt, 1984), as harmonic timbre descriptors. Porres details a group of five "dissonance descriptors", which are organized into two categories: (1) sensory dissonance and (2) harmony (see Table 3.3). The discriminating factor between the two groups is the perceptual nature of the dissonance, either by innate/objective factors (sensory dissonance), or cultural ones (harmony).

ATTIBUTE	DESCRIPTOR
Sensory Dissonance	Sharpness
	Roughness
	Tonalness
Harmony	Fundamental Bass
	Pitch commonality (affinity of tones)

Table 3.3 - Perceptual attributes of musical dissonance according to Terhardt.

The descriptors listed in Table 3.3 under sensory dissonance provide measurements of the following three innate or objective factors that contribute to the perception of dissonance: (1) sharpness (also referred to as brightness), (2) roughness, and (3) tonalness. These descriptors have a particularity that is interesting for the scope of my study, which is the possibility to provide meaningful results for all sounds, independently of their nature.

The group of descriptors under harmony refers to subjective aspects of musical dissonance, which are acquired essentially by cultural factors. The term "musical consonance" is used to denote a number of basic auditory phenomena that govern the perception of tonal music. Therefore, descriptors under harmony only provide meaningful results for pitched sounds, such as those produced by traditional acoustic instruments, thus excluding noise, speech, and sounds alike. Harmony encompasses two descriptors: fundamental bass and pitch commonality.

The applicability of psychoacoustic dissonance models as descriptors in my study serves

primarily to characterize the harmonic timbre of audio signals, but is not limited to this use. The first two sensory dissonance descriptors—sharpness and roughness—are appropriated as harmonic timbre descriptors, and the last—tonalness—was used as a mass descriptor, specifically in the report of the noisiness of an audio signal. Concerning the harmony descriptors, the fundamental bass was used as a complementary descriptor of the mass criteria, and pitch commonality will be used later to model the affinity between collections of sound objects (§ 4.1.2).

The following sections will further detail each of the descriptors that characterize harmonic timbre: (1) brightness, (2) width, (3) roughness, and (4) sensory dissonance. All harmonic timbre descriptors fall under the main category of the description scheme because they can measure properties of all perceivable sounds, and offer a representation of the units with a single numerical value.

3.4.2.1 - Brightness

The brightness of a sound is related to the centroid of its spectrum representation and is expressed by the magnitude of the spectral components of a signal in the high-frequency range (Porres, 2011). Although the root of this descriptor resides in psychoacoustics, one can also find it in Thoresen's (2007b) musicological theory, which pinpoints its importance in linguistics—in order to distinguish between the sounds of vowels and consonants—and in music—as a distinguishable factor to perceive different traditional acoustic instruments.

Brightness is computationally expressed by the barycenter of the spectrum. The computation of brightness assumes the spectrum as a distribution, whose values are the frequencies of the spectrum and the probabilities to observe these values are the normalized amplitudes (Peeters, 2004). Brightness is expressed in Hertz and its range has been limited to the audible range of human hearing, which is roughly from 20 Hz to 20

kHz. The output of the descriptor has been further scaled to the interval [0, 1].

3.4.2.2 – Width

Width expresses the range between the extremities of the sound spectral components. It is considered in all three sound-based theories presented in Chapter 2 by Schaeffer, Smalley and Thoresen. In more empirical terms, I may say that the width characterizes the density, thickness, or richness of the spectrum of a sound.

I should explain in better detail the adoption of the designation "width," because it may lead to some misunderstandings. Thoresen (2007b) adopts a related term—spectral width—to characterize a different characteristic of the spectrum, the mass (called noisiness in the proposed scheme). Although width can be seen as a "satellite" descriptor of the mass or noisiness, the two concepts can offer different characterizations of the spectra. Therefore, the reader should realize the difference between the two concepts and avoid misconceptions between the definition proposed here and the one by Thoresen.

An exact computational model of the width of the spectral components of a sound poses some problems, because the spectral representation of the audio signal may encompass an amount of uncontrollable noise, even if the ideal conditions during the recording stage were met. Instead of considering a solution for this problem, which has been a recurrent topic in literature, I adopted a simpler, yet effective workaround. In order to increase both the robustness and reliability of the value expressed by the descriptor, I adopted the low-level descriptor spectral spread as a representation of the width of the spectrum. The spectral spread measures the dispersion of the spectrum around its centroid. In other words, the spectral spread measures the amount of variation of the values (frequencies), assuming that the probability distribution of the values is the normalized amplitudes. In such a way, it does not take into account the range between the extreme frequencies of the spectrum to express its density. Like brightness, the

output of spectral spread is in units of Hertz.

In order to scale the output of the descriptor to the interval [0, 1] the output values are divided by 6500 (this value corresponds approximately to the maximum value obtained from the analysis of the spectral spread of a large number of sounds).

3.4.2.3 - Sensory Dissonance

The descriptor sensory dissonance expresses innate aspects of human perception that regulate the "pleasantness" of a sonority. Even if the sensory dissonance is regulated by a few psychoacoustic factors, it is expressed in the current framework by its most prominent factor, which is the roughness of a sound. In detail, sensory dissonance describes the beating sensation produced when two frequencies are a critical bandwidth apart, which is approximately one third of an octave in the middle range of human hearing (Terhardt, 1974). The partials of complex tones can also produce a beating sensation when the same conditions are met; that is, when they are a critical bandwidth apart. As a result, the timbre of complex tones can affect our experience of roughness (MacCallum & Eibound, 2008).

The computation of sensory dissonance is done by a Pure Data object coded by Alexandre Porres (2011), which implements Richard Parncutt's roughness measure. The sensory dissonance measure is related to the number of tones or spectral components present in a sonority, as well as its amplitude. The sensory dissonance of two pure tones separated by ¼ of critical bandwidth and amplitude 100 dB is approximately equal to 1. Therefore, one may conclude that the sensory dissonance is dependent on the number of tones and partials of a sonority. This poses a problem, because I want to have the same range in all descriptors, and the system does not know a priori how many notes and partials there are in every sound object. In order to solve this problem and scale the descriptors' output to convey the range [0, 1], I used a scaling factor of 0.050, which

seemed the best estimate for a large set of heterogeneous sound examples.

3.4.2.4 - Harmonic Pitch Class Profile

The harmonic pitch class profile (HPCP) is particularly suitable to represent the pitch content of polyphonic music signals, by mapping the most significant peaks of the spectral distribution to 12 bins, each denoting a note of the equal-tempered scale (pitch classes). Each bin value represents the relative intensity of a frequency range around a particular pitch class, which results from accumulating the 25 highest peaks of the spectrum warped to a single octave.

HPCP is also commonly addressed as chroma vector (Peeters, 2006; Serrà, Gómez, Herrera, & Serra, 2008); however, the adoption of the term HPCP in disfavor of chroma vector is due to its widespread use in musical contexts. The word profile used in the descriptor can also be misleading because the term has been applied in the scheme to denote temporal evolutions; however, HPCP is a matter descriptor, because it provides a single representation of a sound object. The HPCP is stored in two different configurations: (1) the resulting accumulated vector, and (2) a normalized vector to the range [0, 1].

3.4.3 - Criteria of Dynamics

The criteria of dynamics describe the loudness of the audio units in two distinct ways: (1) by a single value that offers a representation of the loudness of the overall duration of the unit, and (2) by a curve denoting the dynamic profile of the unit. The first measure is given by the loudness descriptor and the second representation by the dynamic profile.

3.4.3.1 - Loudness

The loudness descriptor expresses the amplitude of a unit by a single value and is defined by the square root of the sum of the squared sample values, commonly addressed as root-mean-square (RMS). The loudness descriptions are computed by Miller Puckette's object sigmund~, which is included in the software distribution of Pure Data. The output of the object is consequently scaled to convey the interval [0, 1].

If the units have a considerably long duration, the value expressed by the loudness descriptor may be relatively crude, since it is a temporal phenomenon by excellence. However, even if the representation of the units' loudness by a single value may be seen as oversimplifying or too loose for the description of this perceptual phenomenon, it may constitute reliable information for many applications when compared with a full detailed description of the envelope curve over the duration of the unit.

3.4.3.2 - Dynamic Profile

The dynamic profile is a descriptor that belongs to the criteria of form since it represents the evolution of the units' amplitude. The curve is scaled by a factor of 0.001 to convey the interval [0, 1] and is expressed in two different ways: (1) by an amplitude envelope and (2) by basic statistics—such as minimum, maximum, mean, standard deviation, and variation—that represent the amplitude envelope shape (by considering it a probability distribution). Figure 3.1 shows an example of the dynamic profile information extracted from a sound object.

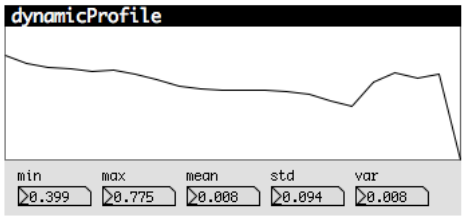


Figure 3.1 - Dynamic profile of a sound object and the values extracted from the profile.

3.5 - Summary

In this chapter, I proposed strategies for segmenting an audio continuum into sound objects using onset detection and beat tracking methods, along with a morphological scheme for describing their most prominent perceptual characteristics. The description scheme results from the interaction between musicological and psychoacoustic theories and MIR research, in particular the literature related with audio descriptors. In addition, the terminology adopted relies on empirical terms borrowed from musical theory and practice in order to increase usability.

The description scheme is divided in three major perceptual criteria—mass, harmonic timbre, and dynamics—that unfold in a set of ten descriptors. The descriptors may also be categorized according to their output representation according to two concepts borrowed from Schaeffer: matter and form.

Audio descriptors under matter are defined in a linear continuum and adopt the same range: the interval [0, 1]. While the interval limits of each descriptor corresponds to specific types of sounds, there is no strict one-to-one correspondence between regions of the interval and specific sound typologies. To achieve such uniformity in the descriptors' range some scaling is applied to the descriptors output. However, the scaling factor is not relative to the maximum, minimum, or mean values of the descriptors functions. Instead, the scaling I use in some descriptors is relative to fixed values determined by specific

perceptual characteristics of the features in question. Therefore, not only is a "normalized" range guaranteed, but also meaningful information concerning the audio signal's content.

Despite the recent tendency to adopt large numbers of audio features in content-based audio processing systems in order to enhance their results, earGram purposefully encompasses a very limited number of descriptors. The adoption of a limited number of descriptors relies on recent studies (Mitrovic, Zeppelzauer, & Eidenberger, 2006; Peeters, Giordano, Susini, Misdariis, & McAdams, 2011), which argue that the information expressed by the totality of audio descriptors developed so far expose a high degree of redundancy. Therefore, I can conclude that my description scheme provides a rich representation of audio signals, since it covers the most significant classes of audio descriptors.

Chapter 4

Musical Patterns

In this chapter, I extend the analytical tools presented in the previous chapter by inspecting higher layers of musical structure. In other words, while the analytical strategies presented in Chapter 3 departed from the most basic representation of audio signals (i.e., the sample), the algorithms detailed here rely on sound objects' descriptions in order to extrapolate representations of the higher temporal scales of musical structure. Ultimately, the aim of this chapter is threefold: (1) to create models of the temporal dynamics of the music (§ 4.1); (2) to discuss and propose strategies to compare and group sound objects (§§ 4.2-4.4); and (3) to provide mid-level descriptions of the corpus (§ 4.5).

The following sections will introduce the reader to algorithms that may ultimately answer the following questions: how common is a particular characteristic throughout the audio source? Which features are more relevant? Which units recur, and in which order? Are there any outliers? How similar are the units in the corpus? Are any representative groups of units within the corpus? How are they organized in time? Does the original

temporal sequence of the units denote the use of repeating patterns?

4.1 - Probabilistic Models of Musical Structure

Given the temporal nature of music, the ability to represent the dynamics of musical structure is at the core of any analysis-synthesis system. Accordingly, earGram adopts strategies to model transitions between sound objects in order to map the dynamics of musical structure. In other words, earGram uses algorithms to learn and encode the temporal evolution of particular audio features of musical structure. In order to do that, I adopt the state-space models named *n*-grams, which encode sequences of discrete events using statistical properties (Jurafsky, Martin, Kehler, Vander Linden, & Ward, 2000). *N*-grams are amongst the most used strategies to encode musical structures computationally (Cont, 2008). They provide a representation of time-indexed sequence of graphs (nodes and edges) where each node refers to a state of the system over time. When dealing with musical elements, the states may represent musical events of different time scales, such as sound objects (e.g. notes, chords, silences), meso structures (e.g. rhythmic patterns, melodic arcs), and macro structures (e.g. sections).

I adopted *n*-grams because they embed a property that is seminal for my framework: they provide the basis for a Markov chain algorithm, which is an algorithm utilized in earGram for generating musical sequences. While the creation of the *n*-gram representations will be examined in the following section, its application for the generation of musical structures will only be addressed in the second part of this dissertation.

The models that will be presented not only learn and encode the dynamics of three elements of the audio source's structure—noisiness, timbre, and harmony—but also "artificially" establish optimal transitions and overlaps between sound objects based on psychoacoustic theory principles. It is important to highlight that the modeling strategies

implemented in earGram only encode singular features of the original audio data because the goal is not to provide a comprehensive representation of all dimensions of musical structure and their inter-relationships, as it is attempted in many style imitation approaches to music (cf. Cope, 1996, 2001). Instead, I adopt models that provide a basis to assist and ease the process of music creation through sampling techniques by automating some of the parameters of a composition.

A detailed explanation of the *n*-grams creation will be presented in the following sections. Section 4.1.1 details models that learn and encode particular elements retrieved from the structure of the audio source(s), and sections 4.1.2 and 4.1.3 detail psychoacoustic-based models for transitioning and superimposing audio objects.

4.1.1 - Modeling Elements of Musical Structure

EarGram creates *n*-grams that encode the temporal dynamics of the following three elements of the audio source(s) structure: noisiness, timbre, and harmony. My software starts by learning the probability of transitioning between discrete elements of musical structure for each of the aforementioned characteristics, and, consequently, stores all probabilities in a matrix. The modeled events need to be discrete features that are extracted from the sound objects, and the temporal dimension of the models encodes the original sequence of units.

The elaboration of transition probability tables is fairly straightforward to compute. However, when dealing with audio signals, to obtain a finite-state space for each modeled element may pose some problems. If the states were directly observable, as in symbolic music representations, no pre-processing would be necessary. However, this is hardly the case when dealing with audio data. Thus, I applied a different strategy in order to create a finite-state space for each of the three musical characteristics.

The noisiness descriptor characterizes the units in a linear continuum, whose limits are

zero and one. Given the need to have a finite number of classes to create a transition probability matrix, the range of the descriptor was arbitrarily divided in ten equal parts. Each class is represented by a numerical value from zero to nine, sequentially distributed in the interval from the lower to the upper limits. Timbre is expressed by a single integer that represents the three highest bark spectral peaks. The algorithm to find the compound value is shown in Table 4.1. Finally, the pitch class of the fundamental bass represents the audio units' pitch/harmonic content.

After obtaining the finite-state space for each characteristic, I computed the creation of a transition probability matrix in the following steps: (1) accumulating the number of observations from the *n* previous states to the following state and, after the totality of the sequence is considered, (2) divide each element of the matrix by the total number of observations in its row. The resulting matrix expresses the probabilities of transitioning between all events.

Operation Number	Operation Description	Example
1	Sort in an ascending order the three peaks with highest magnitude	5, 14, and 15
2	Convert the integers to binary	101, 1110, 1111
3	Shift the 2nd and 3rd numbers by 5 and 10 cases to the left[10]	101, 111000000, 11110000000000
4	Convert the result to decimal	5, 448, and 15360
5	Sum the resulting values	15813

Table 4.1 - Flowchart of the algorithm that reduces the Bark spectrum representation to a single value.

A final note should be addressed to the order of *n*-grams used. By default, it is adopted a third order *n*-gram, that is, the algorithm encodes the probability of transitioning between the three last events and the next one. However, the user can easily change this

[10] This bitwise operation allows the codification of the three values in non-overlapping ranges, which makes the sum of their decimal representation a unique value for any possible combinations of three values that the algorithm can adopt (0-23).

parameter. If the corpus has a considerable number of units, increasing the order of the *n*-gram may enhance the resemblance of the generated output to the original audio. The inverse procedure should be applied to corpora with a very small number of units.

4.1.2 - Establishing Musical Progressions Based on Pitch Commonality

All models exposed in the previous section rely on the structure of the original sequence of the units to formulate the probability of transitioning between musical events. In this section, I present a different strategy to determine the probabilities of transitioning between sound objects, which does not rely on the structure of the audio source(s). Instead of modeling a particular characteristic of the audio source(s) by learning its internal organization, the method presented here defines the probabilities "artificially" by applying a psychoacoustic dissonance model, in particular by computing the pitch commonality between all units.

Pitch commonality provides a link between psychoacoustics and music theory and it is defined as the degree to which two sequential sounds have pitches in common. It measures the "pleasantness"[11] of the transition between two sounds, and can be seen as an oversimplification of harmonic relationships (Porres, 2011). For instance, the pitch commonality of musical intervals is quite pronounced for perfect octaves, less pronounced for perfect fifths and fourths, and more or less negligible for any other intervals.

The computation of pitch commonality depends on the amount of overlapping pitch saliences between two sounds. The pitch salience is defined as the probability of consciously perceiving (or noticing) a given pitch (please refer to Parncutt (1989) for a detailed description of its computation). Pitch commonality is calculated by the Pearson

[11] The concept of pleasantness is understood here as sounds that express a low degree of sensory dissonance (see § 3.4.2.3 for a definition of sensory dissonance).

correlation coefficient[12] of the pitch salience profiles across the frequency spectrum of two sonorities (Porres, 2011). It is equal to one in the case of equal spectra and hypothetically minus one for perfect complementary sonorities. For a complete mathematical description of the model please refer to Parncutt (1989) and Parncutt and Strasburger (1994).

Initially, earGram creates a matrix that stores the results of the pitch commonality calculation between all pairs of units in the corpus. Consequently, all elements of the matrix are converted into probabilities. The last step is done by dividing the absolute value of each element in the matrix by the sum of all absolute values in its respective row. The resulting matrix is the transition probability table of a first-order Markov chain algorithm.

4.1.3 – Vertical Aggregates of Sound Objects Based on Sensory Dissonance

CSS deals primarily with the horizontal dimension of the music, that is, the generation of musical sequences. However, it is current practice to expand the technique to address the synthesis of overlapping units (Schwarz, 2012; Schwarz & Hackbarth, 2012). Despite the popularity of this new approach, the resulting sound quality of the vertical superposition of audio units has been overlooked. So far, there is no consistent method to define the sonic quality of target phrases that encompass vertical aggregates of audio units.

The vertical dimension of music is related to the relationship between simultaneous events, or the sonic matter and its constituent components. According to Thoresen (2007b), the primary structural element of the vertical dimension in Western music is harmony. Timbre can be considered a secondary element. The description scheme

[12] The Pearson correlation coefficient is often used to determine the relationship between two variables by measuring the linear correlation between them. It is calculated by the covariance of the two variables divided by the product of their standard deviations. The Pearson correlation coefficient may adopt values between minus one and one. Zero expresses no association between the two variables, minus one indicates total negative correlation, and one indicates total positive correlation (Taylor, 1990).

presented earlier (§ 3.4) allows the characterization of the vertical dimension of the sound objects, such as the width or degree of sensory dissonance of sound. However, from a creative standpoint, the use of any of these descriptors is confined to the horizontal organization of music. The sensory dissonance descriptor does not express much about the sonic result of simultaneous layers of audio units.

I adopted the sensory dissonance descriptor in order to characterize and organize vertical aggregates of audio units, but in a different manner as used in the description scheme. To measure the "pleasantness" of two simultaneous units, I computed the degree of sensory dissonance between the combination of the spectral representations of the two units (see § 3.4.2.3 for a detailed explanation of the computation of sensory dissonance). A matrix stores the results of the sensory dissonance measures between all pairs of units in the corpus (see Figure 4.1). The resulting matrix will be utilized later to guide the generation of vertical aggregates in earGram.

Unit number	1	2	3	…
1	1	0.1	0.2	
2	0.1	1	0.5	
3	0.2	0.5	1	
…				

Figure 4.1 – Example of a matrix that exposes the sensory dissonance between all pairs of sound objects in the corpus.

Above, I have detailed the creation of five *n*-grams that encode optimal transitions and the superposition of sound objects. The creation of the models relies on descriptions of sound objects, whose computation was presented in Chapter 3. The following sections will continue to examine how musical structure can be apprehended and/or extrapolated, but the focus will shift towards higher layers of musical structure. In order to provide

strategies that ultimately expose the higher layers of musical structure, I will first introduce how sound objects can be consistently compared.

4.2 – Audio Similarity

Sounds can be compared to other sounds according to numerous properties. Tristan Jehan (2005) summarizes the criteria with which we can estimate the similarity between two songs to the following five categories: (1) editorial (title, artist, country), (2) cultural (genre, subjective qualifiers), (3) symbolic (melody, harmony, structure), (4) perceptual (energy, texture, beat), and (5) cognitive (experience, reference). A definite measure of similarity between two songs or audio samples does not exist (Jehan, 2005). Music similarity is an ambiguous task, not only because it is a very complex multi-dimensional problem, with varied subjective dimensions, but also because it is context-dependent. In other words, the evaluation of the similarity between audio signals is highly dependent on the application context and the user.

A systematic computational model of music similarity poses even more problems, which can be roughly summarized in three topics: (1) the complexity of the task, (2) the subjectivity of criteria, and finally, (3) the difficulty of algorithmically considering and/or encoding application contexts and users' preferences. While addressing computational models of similarity, it should be noted that I do not refer to any work that operates on metadata supplied by humans, or even high-level music representations, such as MIDI. Of interest here is the measurement of similarity between non-uniform audio units, generally ranging from a fraction of a second to a few seconds, which are represented by an audio feature vector. Ultimately, the aim of discussing audio similarity measures in the context of this dissertation is to provide a reliable method for comparing and depicting audio units according to their similarity.

In order to compute the similarity between audio samples one usually calculates the

distance between their representative feature vectors. The choice of the audio descriptors that compose the feature vectors will determine the quality of the computed similarity, which should account for not only the nature of the signals being compared, but also the context of its application. In most content-based audio systems, this particular task is commonly hard to achieve because most descriptors involved in audio similarity computation are meaningless for most people. Therefore, one cannot expect users to be able to define and restrict the set of features to convey specific needs.

The description scheme I utilize in earGram minimizes the drawbacks of this operation because the descriptors are already adapted to musical imperatives and adopt terminology from music theory and practice. In addition, the use of a standard scale for all descriptors not only prevents some unbalanced comparisons, resulting from disparities in the descriptors' range, but also avoids the necessity of normalizing the descriptors' output, which consequently allows the preservation of meaningful information about the audio units' characteristics according to specific sound typologies.

The description scheme proposed earlier also embeds a characteristic that is critical to achieve reliable comparisons between non-uniform audio units: the audio descriptions used are invariable to the audio units' duration. The similarity computation between audio units with different durations raises some problems because the results of most descriptors depend on the duration of the analyzed units (Goto, 2003; Foote, 1999; Ong & Herrera, 2005; Jehan, 2005; Paulus, Müller, & Klapuri, 2010). In order to provide reliable comparisons between non-uniform audio units, the proposed description scheme only examines audio features that are invariant to the units' duration, particularly because their computation relies essentially on ratios between spectral components.

Finally, I will detail the strategies implemented in earGram to assign weights to the audio features at issue in order to enhance the quality of the results towards specific applications and uses. The weights in the context of the current study may bias the similarity measure towards specific properties of the audio source(s) one wants to

explore. For example, if we want to segregate audio units according to different pitched instruments that have distinct ranges, one may increase the weight of descriptors such as brightness and pitch.

Despite the considerable body of knowledge on the automatic assignment of weights to descriptors in concatenative TTS synthesis (Hunt & Black, 1996; Macon, Cronk, & Wouters, 1998; Meron, 1999; Lannes, 2005), in CSS the different nature of the music signals poses problems that are not yet solved. Assigning weights to descriptors involved in music similarity computation is rather difficult (Sturm, 2006b). In most CSS systems, the procedure is done manually and very little guidance is provided. In addition, common descriptors used in CSS (low-level audio features) are meaningless for most users, which leads to the assignment of arbitrary weights that may give rise to inconsistent results.

Some attempts have been made to devise automatic strategies to assign weights to audio descriptors according to the characteristics of the corpus. Brent (2009) ascribes weights according to the variance of the descriptors, and Norowi and Miranda (2011) proposed the use of analysis hierarchy processes, a promising semi-automatic technique, to prioritize audio features based on user input that rates the degree of importance of each feature by comparing it to all remaining features used in the system. I adopt three strategies to specify the weights of audio features: (1) manual, (2) automatic specification according to the variance of the audio features, and (3) manual restriction and prioritization of audio features.

The manual assignment of weights to audio features can be very simple and effective if the user has a working knowledge of the audio source(s) and/or descriptors available. EarGram minimizes the latter barrier because the set of descriptors devised already adopts terminology from music theory and practice, which is by far more accessible than MIR jargon. In addition, earGram provides a strategy to visualize multidimensional feature vectors that offers a clear representation of the distribution of the data for each descriptor, which can contribute significantly to understanding the most distinguishable

features of the corpus. The algorithm behind the visualization is named parallel coordinates (Inselberg, 2009). The visualization is done on top of parallel lines, commonly placed in a vertical position and equally spaced. A point in *n*-dimensional space is represented as a polyline whose endpoints for each segment (vertices) fall on the parallel axes. The position of the endpoints on the i^{th} axis corresponds to the i^{th} value of the feature vector (see Figure 4.2).

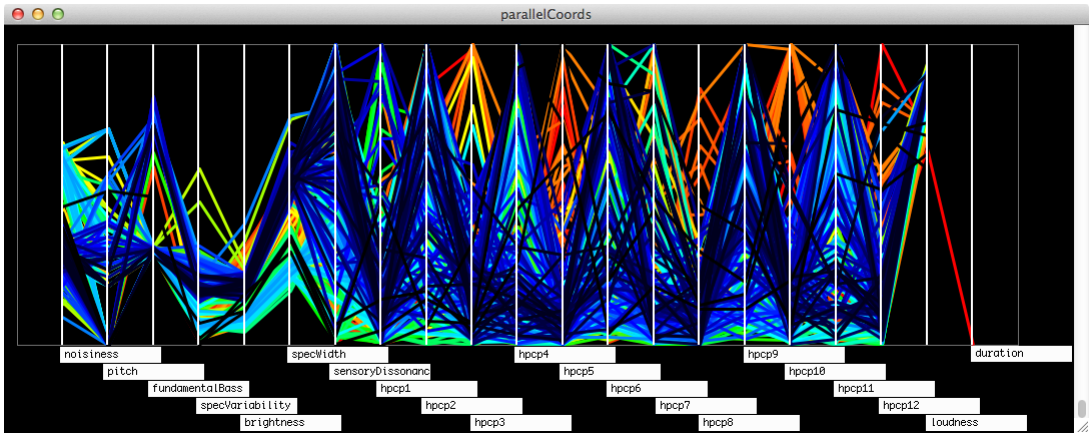


Figure 4.2 – EarGram's parallels coordinates visualization of a corpus comprising a single audio track—*4* by Aphex Twin.

The weights may also be automatically assigned according to the variance of the features. Weighting features according to their variance assumes that features with higher variance enhance the similarity computation, because they provide a more distinctive characterization of sound objects. If the user does not want to manually specify and experiment with different weights, this strategy may be the ideal choice, because it is automatic and clearly enhances the distinction between the sound objects present in the corpus.

The third strategy, the specification of feature sub-collections and their prioritization in the unit selection process is particularly relevant here and is an adopted strategy in the recombination methods detailed later in Chapter 6. The need to restrict the proposed description scheme to a smaller set of audio features relates to the different nature of the audio source(s). For example, when dealing with soundscapes, the use of the pitch

descriptor to guide the recombination of audio units may provide some erroneous information and should be used carefully. The prioritization of features is seen here as a constraint-solving problem. The constraints are defined hierarchically, which reduces the corpus to sub-spaces that optimally satisfy all constraints at runtime. The application of this strategy is particularly detailed in section 6.4.1.

After this discussion on strategies of, and problems raised by, the comparison between individual sound objects, I will expand the perspective on music similarity to include groups of sound objects that share perceptual features. In order to reveal such groupings, I adopted three clustering algorithms, whose conceptual and technical details are discussed next.

4.3 – Clustering

Clustering, or cluster analysis, is the process of grouping sets of objects that present common characteristics. Clustering is frequently applied to solve or assist various problems in machine learning, pattern recognition, image analysis, and information retrieval.

For earGram, I implemented three clustering algorithms with the aim of grouping sound objects with similar features. Ultimately, the resulting clusters reveal musical patterns that can be utilized differently during performance. The current implementation comprises three non-hierarchical clustering algorithms: (1) *k*-means, (2) quality-threshold (QT) clustering, and (3) density-based clustering algorithm (DBSCAN). I chose this set of algorithms because they rely on different parameters defined in advance, and offer very different clustering configurations. The parameters defined a priori for each of the algorithms have direct implications in the resulting shapes of the clusters and can be associated with musical parameters.

If the user wants to have a concise number of clusters defined a priori and consider all sound units the choice should fall on *k*-means. On the other hand, if the user wants to define clusters based on parameters such as similarity thresholds or neighborhood proximity between units, he/she should choose either QT-clustering or DBSCAN, respectively.

Before detailing the technical implementations of the clustering algorithms, I need to address some remarks regarding the dimensionality of the feature vectors considered by all clustering algorithms. Even though the clustering algorithms can deal with arbitrary long vectors, in order to convey a clearer and more understandable visualization of the results and allow physical navigation, I restricted the computation to two-dimensional feature vectors. Therefore, the sound objects are represented as two-dimensional data points in a plane prior to the clustering. The predominant use of two-dimensional vectors over multidimensional vectors will be further justified and discussed in a coming section (§ 4.4.1).

4.3.1 - *K*-means

K-means is one of most popular clustering algorithms. It partitions a collection of data points into *k* clusters (defined a priori) by allocating each point to the cluster with the nearest centroid (MacKay, 2003). The main purpose behind the implementation of *k*-means in earGram is the possibility to divide the collection of sound objects into an exact number of representative clusters. For example, if the user wants to generate a specific number of concurrent layers of units, he/she may want to adopt *k*-means to create a sub-corpus of units that somehow share particular characteristics.

K-means is an iterative algorithm that at each run assigns all data points to their nearest centroid. At the end of each iteration, the centroids' positions are reallocated in

according to the points that were assigned to the cluster they represent. When the algorithm reaches a stable configuration, the iteration process stops (MacKay, 2003).

The implementation of *k*-means does not guarantee that it will converge to the global optimum, because it is a heuristic algorithm. The results will be highly dependent on the initial position of the centroids. Therefore, in earGram, the initial centroid positions are assigned to the collection of *k* points that give the minimum distortion[13] from a collection of 50 arbitrary sets of centroid positions. Although the described initialization has some computational costs associated, it will provide a faster computation of the results and hopefully the convergence to the global optimal configuration.

4.3.2 - Quality-Threshold Clustering

QT-clustering is an algorithm that was first presented by Heyer, Kruglyak, and Yooseph (1999) as a strategy to cluster gene expression patterns. The primary characteristic of this clustering algorithm is the possibility to specify a quality threshold and the minimum number of data points per cluster. In earGram, QT-clustering may be useful to distinguish and consequently group different classes of sounds, such as sounds produced by different instruments, as well as to define aggregates of sounds objects, which do not exceed a similarity threshold.

Relying on the user-assigned threshold distance between data points and the minimum number of sound units per cluster, it is possible to detail the algorithm in five steps: (1) compute candidate clusters for each data point in the corpus by including for each candidate all reachable points within the distance threshold, (2) store the candidate cluster with the most points, (3) check if the candidate cluster meets the minimum number of data points, per cluster, defined in advance (4) remove all the points of the

[13] The distortion is calculated by the sum of the squared distances between each data point and its allocated centroid.

stored cluster from further consideration, and finally (5) repeat the operations from the first point with the reduced set of points until no more clusters can be formed.

QT-clustering considers all possible clusters. The candidate clusters are generated with respect to every data point and tested in order of size against the quality criteria. Major advantages of this clustering technique are the detection of outliers that can be treated differently at runtime (for instance, excluded from the recombination) and the precise control over the similarity of grouped units. The major disadvantage of this clustering strategy is its heavy computational cost.

4.3.3 - Density-Based Clustering

DBSCAN is a well-suited algorithm to discover clusters of arbitrary shapes in spatial databases. Clusters are defined according to two parameters: (1) the distance threshold or neighborhood proximity between data points and (2) their density (the minimum number of points within the radius of each unit to form a cluster). In other words, each unit in the cluster must not exceed an assigned distance threshold from another unit in the cluster and each cluster has to contain at least the specified minimum number of units within the distance threshold. Therefore, the formed clusters have a typical density of points, which is considerably higher than outside of the cluster.

My implementation of the DBSCAN algorithm follows Ester, Kriegel, Sander, and Xu (1996). EarGram starts by inspecting an arbitrary unit that has not been visited by the algorithm. This unit's neighborhood is examined, and if it includes enough points within the threshold distance, a cluster is defined. Otherwise, the point is labeled as "noise." However, this unit might later be found in the neighborhood proximity of a different point. If a unit is found to be a dense part of a cluster, all of its neighborhood units (within the distance threshold) are also part of that cluster. This process continues until the density-connected cluster is found and repeated until all units have been visited.

Similarly to the QT-clustering algorithm, DBSCAN avoids defining a priori the number of clusters. However, the algorithm finds arbitrarily shaped clusters that are very different from the ones found by the QT-clustering. It can even find clusters surrounded by (but not connected to) a different cluster. DBSCAN helps define action zones or large groups of interconnected data points according to a proximity distance and density, which may sort and group sound objects by “scales.” The resulting clusters will encompass sound units that express some continuity, that is, each cluster exposes perceptual trajectories of particular audio features. Similarly to QT-clustering, DBSCAN detects outliers, which may help restrict the corpus to a more unified collection of units during performance.

The audio similarity and clustering algorithms detailed in the last sections are better understood through visuals. Corpus visualizations were adopted in earGram not only to expose the hidden results concerning audio similarity and clustering, but also to provide tools in which one can navigate, explore, and interact with the corpus.

4.4 – Visualizations

EarGram adopts two visualization strategies: 2D-plot and similarity matrix. Both allow the navigation, exploration, and interaction with the corpus, and also aim at depicting various (hidden) analytical stages of the system. Ultimately, the implemented visualization strategies reveal some intrinsic characteristics of the audio source(s), in particular its macrostructure by depicting the similarity between the sound objects that compose the corpus. The visualizations may assist in the decision-making processes during performance. In addition, they also allow interactive and guided explorations of the corpus. A detailed description of the two visualization strategies, along with their technical implementation, follows.

4.4.1 – Sound-Space

Sound-space offers a visual representation of the sound objects' collection in a 2D-plot. It provides an intuitive visualization of the similarity between sound objects and is particularly suitable for browsing and exploring a corpus of audio units by navigating through its representation. 2D-plots are one of the most common visualizations adopted in CSS, and frequently allow physical interaction with the corpus of audio units (Martin, 2011; Schwarz, 2012; Schwarz & Hackbarth, 2012).

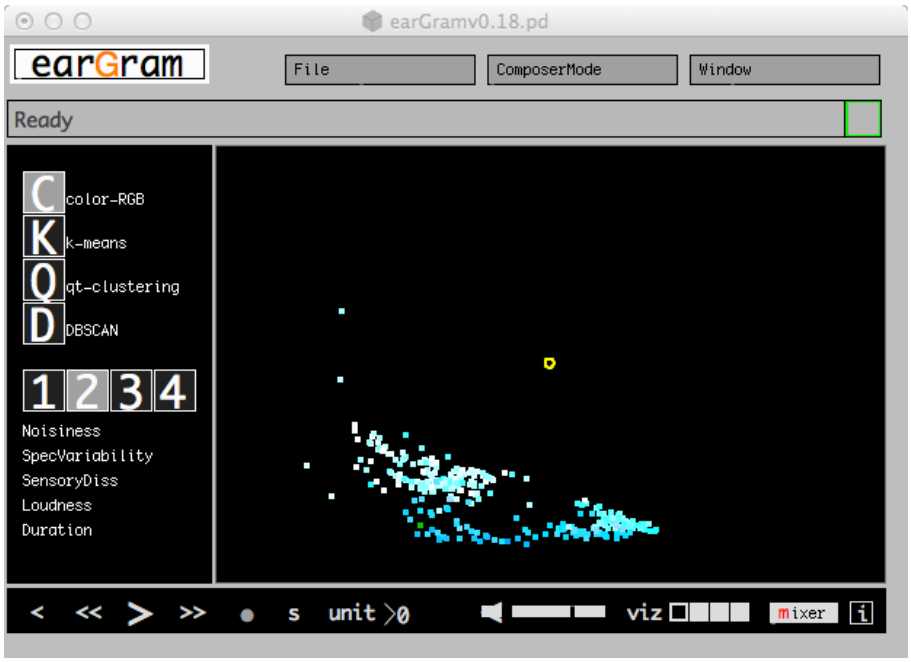


Figure 4.3 - Visual representation of a corpus of audio units comprising a single audio source—*4* by Aphex Twin—in a 2D-plot whose axes were assigned to the following descriptors: noisiness (*x*-axis) and spectral variability (*y*-axis). The units' color is defined by sensory dissonance, loudness, and duration, by assigning each descriptor to the values of R, G, and B, respectively, and using an additive color model.

The following sections detail two different approaches to constructing the sound-space visualization in earGram. The first assigns individual audio features to the axes of the 2D-plot, and the second adopts multiple audio descriptions in the representation, which can

be depicted in two dimensions with the help of multidimensional reduction algorithms. A common element to both approaches is the layer of information that is offered by the units' color. The color of each unit in sound-space is defined by a list with three elements that correspond to the red, green, and blue values of an additive RGB color model. The R, G, and B values represent audio features from the available set of descriptors.

4.4.1.1 – Two-Dimensional Visualizations of the Corpus Using Binary Sets of Descriptors

In sound-space, the use of different binary sets of audio features to depict a corpus of audio units provides valuable information about the corpus, in particular the similarity between its constituent units. In order to create such visualization of the corpus, the user must first assign single audio features to each axis of sound-space. Then, earGram collects the analyzed information of both selected features for all audio units in the corpus, and depicts the corpus according to those values. In other words, the coordinates of each audio unit (single dot) in sound-space are two values (*x* and *y*) that correspond to audio features of the units. Therefore, the sound-space visualization provides as many visualizations of the corpus as the number of possible combinations between all pairs of audio features provided by earGram's description scheme.

In addition to the depicted information, the sound-space visualization also allows the definition of target phrases to be synthesized with a high level of precision by navigating its representation. In fact, sound-space functions in a similar way as traditional acoustic instruments. Its response is predictable, direct, and controlled, because the audio units are represented in a scale whose limits are specific types of sound. If the same conditions are met, in particular the same pair of descriptors, the same performance gestures result in the same sonic response. An important distinction is the mutability of the instrument according to the feature space of the 2D-plot. Changing the feature space in sound-space can be seen as changing a preset in a synthesizer. The adoption of different sets of

descriptors to depict the corpus of audio units imposes significant changes in the sonic feedback. If one repeats the same trajectory in sound-space with different sets of descriptors the sonic feedback of the gestures can be regarded as variations. Each compound set of descriptors has its own identity.

A two-dimensional representation of the sound objects according to the descriptors pitch and loudness is a clear example of how the sound-space visualization may emulate a piano-keyboard behavior. However, earGram was not designed to primarily emulate this behavior. Instead, its purpose focuses rather on the exploration of all aspects of sound outside the pitch-duration primacy. For example, the combinations of pitch or fundamental bass descriptors with any of the harmonic timbre descriptors (e.g. brightness, width, and sensory dissonance) provide an extended control over the harmonic quality of the pitch/chords. One thus may “modulate” the timbre of particular pitches.

The use of noisiness, brightness, width, and sensory dissonance is particularly effective for visualizing a corpus of electronic-generated sound units. This is due to the fact that these types of sounds commonly expose a rich variety of colors (timbre). Finally, any combination of the following descriptors: noisiness, loudness, width, and spectral variability is particularly interesting to visualize and control the synthesis of environmental sounds.

I should also remind the reader that the descriptors pitch and fundamental bass do not provide meaningful information to all types of audio units. For this reason they were presented as complementary descriptors (see Table 3.2). To conclude, I just would like to add that the creation of sound events/structures organized by parameters other than pitch, duration, and loudness are not very common in Western music and their use should undergo an experimental phase that goes beyond the temporal scope of this investigation.

4.4.1.2 - Multidimensional Reduction of the Descriptor Space

The most common way to describe and represent units in content-based audio processing applications is to include a large number of audio features in a multidimensional vector. In recent years, a large number of new descriptors have been presented and adopted despite the higher computational costs. We should also keep in mind that the incorporation of a larger number of descriptors does not always represent an improvement in the characterization of the audio signals.

High-dimensional vectors are difficult to visualize and are not at all suitable for physical navigation, which is commonly performed in two- or three-dimensional spaces. Hence, the most common solution to this problem is either assigning a single descriptor to each of the axis of the two or three-dimensions representation (as discussed in the previous section), or employing dimensionality reduction algorithms to decrease the number of dimensions to two- or three-dimensions while retaining most of the information provided by the vectors. In this section, I propose a dimensionality reduction algorithm to decrease the number of dimensions of the feature vectors, and examine its implication in creative practices, such as musical composition.

Multidimensional reduction techniques are commonly used in content-based audio processing applications. These techniques not only reduce computational costs associated with the matching process, but they also convey the visualization of the corpus in two- or three-dimensions. Hence, the topic could be examined in the current section with respect to the visualization of the corpus or in a different section where I address audio similarity computation. The decision to address multidimensional reduction techniques here is for a very simple reason: while in earGram the visualization of high-dimensional feature vectors in a 2D-plot is a very a pertinent problem, the description scheme does not encompass a very large number of descriptors and therefore does not pose tremendous problems in

terms of computational cost.

I noted in the direct contact with a few composers that utilized earGram in their compositions that dimensionality reduction methods raised curiosity, even if the outcome of the algorithm is slightly misleading. The visual representations gathered after applying multidimensional reduction algorithms lack clarity. The axes of the plane are hardly related to any particular feature and what remains is a general and "blind" representation of similarity between the sound objects.

The precise manipulation of individual dimensions of the sound matter is the most common approach in musical composition. The same level of accuracy is achieved if single features are assigned to each of the axis of a plane. However, dimensionality reduction methods might be helpful in cases where it is unclear which features the user wants to control, or when it appears that no pair of features will provide satisfactory results. The latter case might be true for applications using sounds of very different natures.

Two of the most popular algorithms for dimensionality reduction are principal component analysis (PCA) (Shlens, 2005; Skočaj, Leonardis, & Bischof, 2007) and multidimensional scaling (MDS) (Mikula, 2008; Schwarz & Schnell, 2009). Both methods can be used to obtain smaller representations of high-dimensional feature spaces.

EarGram adopts the algorithm star coordinates (Kandogan, 2000) for dimensionality reduction. The algorithm is substantially less known and applied than PCA or MDS, especially to address audio feature vectors. However, star coordinates offers two major advantages over the aforementioned algorithms: (1) the understandability of the axes after processing takes place, and (2) its suitability for both online and offline processing. A clear disadvantage of star coordinates is the need to explore the representation by weighing the variables and assigning different angles to the axes to find interesting patterns.

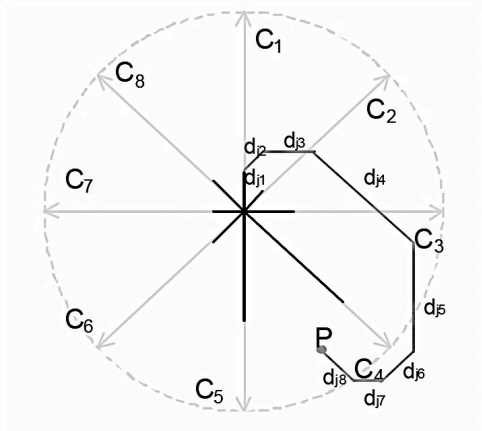


Figure 4.4 - Mapping of an eight-dimensional point to two dimensions. Axes are named as C_x, each dimension of the point as d_{jx}, and P is the final point position (Kandogan, 2000). (Copyright 2000 by Eser Kandogan. Reproduced with permission.)

Star coordinates map a high-dimensional point linearly to two dimensions by summing the vectors resulting from the point coordinates arranged on a circle on a two-dimensional plane with equal (initial) angles between the axes with an origin at the center of the circle (see Figure 4.4 for a demonstration of the algorithm).

4.4.2 - Self-Similarity Matrix

By depicting pairwise similarity between the same original sequence of sound objects assigned to both vertical and horizontal axes of a square matrix, it is possible to reveal patterns of the audio source(s) that ultimately expose the macro structure of the data. The graphical representation is called a self-similarity matrix. The technique was first introduced by Foote (1999) with the aim of visualizing musical structures. The method consists of building a square matrix where time runs from left to right, as well as from bottom to top, and the cells' color denotes the similarity between audio units. The similarity between sound objects is computed by the Euclidian distance between their

representative feature vectors. The feature vectors may include a variable number of features from the available set of descriptors, as well as variable weights.

The standard approach to audio similarity matrices consists of computing and depicting the similarity between short windows of fixed duration. In earGram, the compared units correspond to sound objects with non-uniform duration. I adopt similarity matrices in earGram mainly to guide the user through the selection of sub-spaces of the corpus that can be used differently during performance.

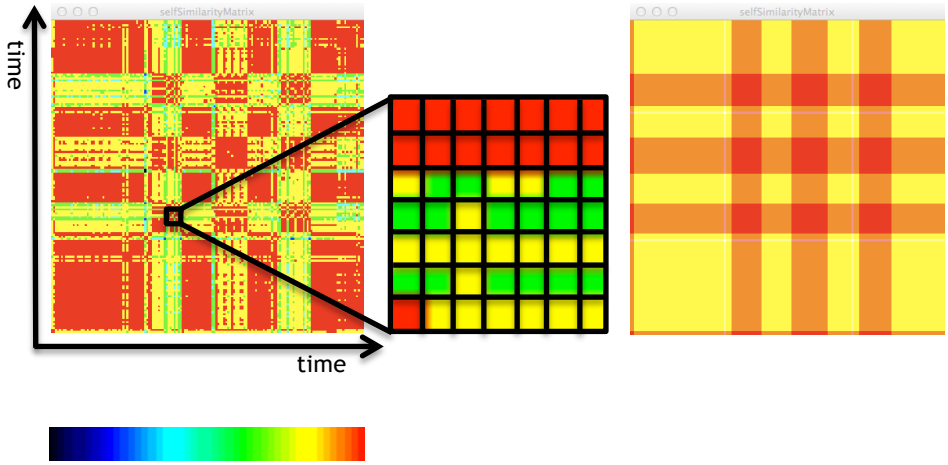


Figure 4.5 – Visualizations of a corpus comprising a single audio track—*4* by Aphex Twin—by a self-similarity matrix (left image) and a related visualization whose color of each cell results from two found clusters in the corpus (rightmost image). The middle image is a detail of the self-similarity matrix, which exposes with detail the color of each cell.

EarGram provides a different visualization strategy of the corpus that follows the same principles behind similarity matrices. The major difference resides in the color of each matrix cell, which instead of resulting from the distance between feature vectors is attributed to the juxtaposition of the audio units' colors (resulting from the clustering strategies). The visualization provides very similar information as the traditional approaches to similarity matrices and enhances its clarity in a similar fashion as reducing

the noisiness of a signal by a smoothing function.

It is important to note that the matrix configurations are highly dependent on the features used to compute the similarity or the clusters. Therefore, in order to find interesting patterns the user may need to explore different collections of features or assign different weights to each descriptor. In addition, both visualizations can provide interesting feedback to the user in understanding how different audio features' weights and constraints, and/or the use of different feature spaces, can alter the notion of similarity between sound objects and their consequent grouping.

Now that I have addressed how sound objects can be consistently compared and grouped to expose characteristics of the higher layers of musical structure, I will conclude the current chapter by providing two descriptions—key and meter—of the corpus/audio source(s) that will be preponderant for some decisions during the composition phase.

4.5 - Mid-Level Description of the Corpus

In this section, I will detail the adoption of two mid-level descriptors for meter and key induction. Unlike the audio descriptors proposed in Chapter 3, the two following descriptors do not target individual sound objects. Instead, they characterize collections of sound objects and may be applied as a strategy to constrain the corpus to sub-spaces of units, or simply to provide information that can be used at later stages of the system for the generation of new unit sequences. The conceptual and technical considerations that assisted the implementation of both descriptors follow.

4.5.1 - Meter Induction

In music, meter refers to the hierarchical organization of time based on perceived temporal regularities (Lerdhal & Jakendorff, 1983). It consists of a periodic system of

stressed downbeats, commonly subdivided on either two (duple meter) or three (triple meter) beats, or any of its multiples.

Great strides have been made to create computational models to induce the meter of a particular music, because it offers valuable information for many musical applications such as music transcription (Schloss, 1985; Klapuri, 2003), editing (Chafe, Mont-Reinaud, & Rush, 1982), and interactive music systems (Malloch, 2005). However, despite all of these efforts, state-of-the art algorithms still pale in comparison to the level of accuracy achieved by humans (Sell, 2010).

One of the most common approaches to meter induction is to find periodic recurrences of musical events, whose first beat (downbeat) is slightly stressed (Cooper & Meyer, 1960; Gouyon & Herrera, 2003). An important distinction should be made between literature for meter induction algorithms that process discrete musical events (Longuet-Higgins & Lee, 1982; Lerdahl & Jackendoff, 1983; Povel & Essens, 1985; Lee, 1991) and audio data (Goto & Muraoka, 1998; Scheirer, 1998; Gouyon & Herrera, 2003; Klapuri, Eronen, & Astola, 2006; Davis & Plumbey, 2006).

The meter induction strategy implemented in earGram is largely based on Gouyon and Herrera (2003). In brief, the Gouyon and Herrera meter induction algorithm attempts to find regularities in feature vectors sequences through autocorrelation.[14] The resulting peaks from the autocorrelation function indicate lags for which a given feature reveals periodicities. The audio feature over which all processing is done is spectral variability. It is considered the beat as the relevant temporal resolution to extract the features of the audio. Therefore, the aforementioned method is only applied when the audio is previously segmented on found beats. The autocorrelation function examines periods from 2 to 12 beats, and picks the highest value of the accumulated autocorrelation function.

The implemented meter induction algorithm only attempts to find the number of

[14] Autocorrelation is the "correlation of a signal with itself at various time delays" (Dunn, 2005, p. 459). In other words, autocorrelation measures the degree of similarity between a given time series and a lagged version of itself over successive time intervals.

pulses per measure that expose regularities over time. It does not attempt to track the position of the downbeats because the only purpose behind its computation is to find temporal recurrences in the description function. The resulting patterns provide valuable information for guiding generative music strategies, in particular to preserve intrinsic rhythmic features of the audio source(s). In fact, even if the algorithm reports a multiple of the actual meter, it does not disturb the output quality of the generative music algorithms. In addition to the meter, earGram also attempts to infer the key of the audio source(s), another important music-theoretical concept.

4.5.2 - Key Induction

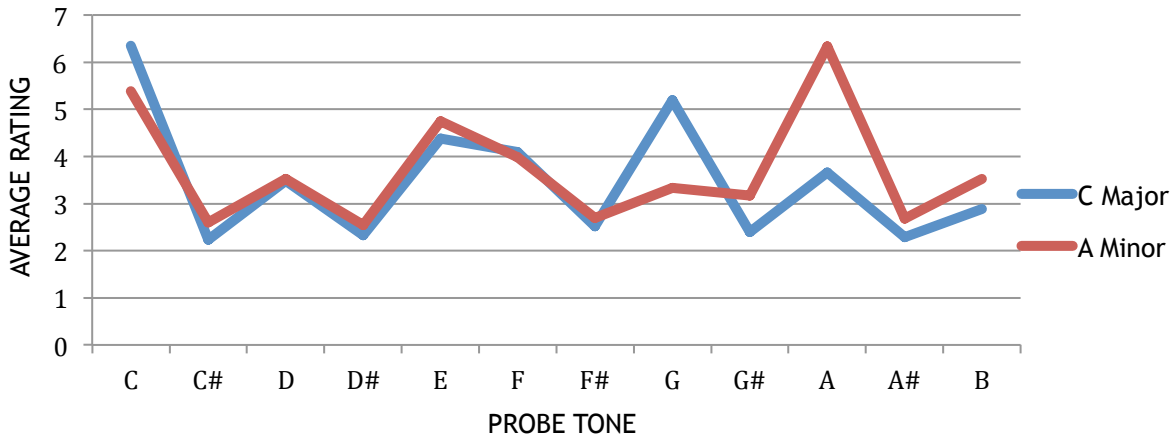
The key or tonality of a musical piece is an important theoretical construct that not only specifies the tonal center of the music, but also hierarchical pitch and harmonic relationships. The tonal system prevalent in the most Western music practices is defined by two elements: a pitch class and a mode. The pitch class corresponds to one of the 12 notes of the chromatic scale, and the mode may be major or minor.

Although determining the tonal center of a musical piece is a rather difficult task for humans, to identify the mode of the key is often intuitive for a human listener. An effective computational model for key induction with the same level of accuracy as a trained musician has not yet been fully achieved (Sell, 2010).

The key induction algorithm employed in earGram is an extension of one of the most prominent and applied algorithms for key induction, the Krumhansl-Schmuckler (K-S) algorithm. Besides its easy implementation and low-computational cost at runtime, the K-S algorithm is quite reliable and effective. Briefly, the algorithm assumes that particular notes are played more than others in a given key. Although the postulate seems pretty evident from a music theory viewpoint, Krumhansl and Kessler have validated the assumption by perceptual experiments (Krumhansl, 1990).

In detail, the K-S algorithm estimates the key of a musical piece by finding the highest correlation between 24 profiles, each corresponding to one of the major and minor scales of the 12 chromatic notes of an equal-tempered scale, and the frequency distribution of the pitch information of a musical piece in 12 pitch classes. The profiles for each of the major and minor scales were devised by Krumhansl and Kesseler in 1990 and are commonly addressed as K-K profiles. The K-K profiles resulted from listening tests, which aimed at finding how well the total chromatic notes of the tempered scale perceptually fit in with musical elements designed to establish a key, such as scales, chords, or cadences. Two major profiles derived form the listening tests, one for the major scales and another for the minor scales, which can be shifted 11 times in order to map the profile to a different tonic. Figure 4.6 depicts the key profiles for the C major and A minor keys.

Some authors have further Krumhansl and Kesseler's research and proposed slight changes to the K-K profiles (Temperley, 1999, 2005; Chai, 2005). Please refer to Gómez (2006b) for a comprehensive review of key profiles used for key induction.



6.35	2.23	3.48	2.33	4.38	4.09	2.52	5.19	2.39	3.66	2.29	2.88	C Major
5.38	2.60	3.53	2.54	4.75	3.98	2.69	3.34	3.17	6.33	2.68	3.52	A Minor

Figure 4.6 - The Krumhansl and Kessler key profiles for C major and A minor keys (Krumhansl, 1990).

It should be noted that the aforementioned algorithm was proposed and extensively applied for the key induction of symbolic music data. Addressing the problem in the audio domain poses different problems, in particular to create the input vector that represents the frequency distribution of pitch classes, because audio data does not provide clean information of the pitch content.

When dealing with audio signals, instead of creating a histogram that accumulates the pitch classes of the audio file, I adopt a vector that expresses the accumulated harmonic pitch class profiles (HPCP) of various frames of the audio as used in related research (Gómez & Bonada, 2005).[15] The accumulated HPCPs express the frequency distribution of the harmonic content of audio signals in 12 pitch classes, and is computed in earGram by wrapping the highest 25 peaks of the audio spectrum into 12 pitch classes. In sum, the key induction algorithm proposed here compares the normalized accumulated HPCP with the K-K profiles in order to determine the most probable key.

It is important to note that there is a disconnection between the representation of the K-K profiles and the HPCP input vector, because the K-K profiles do not consider the harmonic frequencies present in audio signals. Still, they are the most commonly used profiles in state-of-the-art audio key induction algorithms (Purwins, Blankertz, & Overmayer, 2000; Pauws, 2004; Gómez, 2005, 2006a).

Izmirli (2005) and Gómez and Bonada (2005) have addressed the issue of the misrepresentation of harmonic frequencies in the key profiles and offered a similar solution, that is, the creation of harmonic templates obtained by inspecting the harmonic characteristics of a corpus of audio samples and merging it with the key profiles. However, in order to create reliable harmonic templates one should use a similar corpus of sounds as the analyzed musical pieces, reclaiming thus the computation of the profiles each time a different audio source is used. For this reason, the solution I refer to has not been considered in earGram.

[15] For a detailed definition of HPCP please refer to section 3.4.2.4

The key induction algorithm provides valuable information concerning the corpus that facilitates the interaction between different corpora of audio units, or even between the system and a live musician that can be playing along with generated sonic material. In addition, knowing the key of the audio source(s) allows the user to transpose the generated output to any other tonality.

4.6 - Part I Conclusion

Chapter 2 discussed musicological literature that approaches sound description from a phenomenological standpoint, that is, descriptions that focus on the morphology of sounds disregarding sources and causes. My review focused on criteria for the morphological description of sound objects presented by the three following authors: Pierre Schaeffer, Denis Smalley, and Lasse Thoresen.

Chapter 3 relied on the concluding remarks of the aforementioned discussion to formulate musician-friendly computational strategies to segment an audio stream into sound objects and describe their most prominent characteristics. The computational implementation of the scheme, and particularly the segmentation strategies of an audio stream into sound objects also relied on MIR research, in particular literature related to audio descriptors, onset detection, and audio beat tracking. The description scheme implemented in earGram encompasses two main properties that are seminal for this study in particular to the comparison and manipulation of sound objects in creative contexts. The first is the adoption of a limited number of descriptors, which cover the most prominent characteristics of sound and expose low levels of information redundancy. The second is the use of a standard range in all descriptors whose limits are fixed sound typologies; thus avoiding the normalization of the descriptor's output by spectral features and providing reliable information concerning the unit's content in relation to the descriptor's limits.

Chapter 4 proposed computational strategies for modeling the temporal structure of the audio source(s) by establishing the probability of transitioning between all sound objects that comprise the corpus, along with reliable strategies for comparing and clustering audio units, with the ultimate goal of revealing the higher-level organization of the audio source(s). The similarity and grouping are better understood in earGram through the aid of two visualization strategies: 2D-plot and self-similarity matrix. Finally, I provide two algorithms to infer the presence of both a stable meter and key in audio source(s). The purpose behind the analytical strategies devised is either automatic music generation or assisting the composition process, which is addressed in the second part of this dissertation.

PART II: COMPOSITION

Any text is constructed as a mosaic of quotations;

any text is the absorption and transformation of another.

— *The Kristeva Reader*, Julia Kristeva (1986)

The aim of Part II is to explore CAAC strategies that automatically recombine audio units by manipulating descriptions of sound objects as well as to suggest methods for incorporating the generative algorithms in a composition workflow. Part II will adopt a similar, but inverse, structure as Part I. In other words, while the first part of this dissertation adopts a bottom-up strategy for analyzing audio data, the second part adopts a top-down approach to algorithmic composition.

I will adapt well-known CAAC strategies attached to symbolic music representations to address audio signals and function as unit selection algorithms in CSS. My generative methods were implemented and tested in earGram and are able to build arbitrarily long structures in a way that the synthesized musical output reflects some of the elements that constitute the audio source(s). Yet, due to the particularities of my generative methods, the created music is new and different from the raw material that supports its creation—and any other existing music. EarGram demands little guidance from the user to achieve coherent musical results and it is suitable for a variety of music situations spanning from installations to concert music.

Chapter 5

Organizing Sound

Using sound as raw material for a composition is a central concern in electroacoustic music. The simplest approach to compose with sounds in order to create a new composition is by manually manipulating and assembling pre-recorded audio samples. I embrace this method through the recombination of sound objects. However, the recombination process is semi or fully automated by organizing prominent features inferred from the sound objects. The following subsections provide an overview of the technical and conceptual background of the framework's generative component proposed in this dissertation, in order to place it in a particular historical context and justify its pertinence. The chapter concludes by explaining the articulation between the two major modules of the framework: analysis and composition.

More specifically, this chapter provides an historical perspective of sample-based synthesis techniques—sampling, micromontage, and granular synthesis—which contributed to the emergence of CSS. Next, I provide an overview of musical applications of CSS over the last decade. Then, I examine the technical aspects of the framework by asking how

they influence the practice of music composition. The following three compositional approaches will be addressed: (1) the use of sound structure, namely its timbral qualities as the primary material for structuring musical processes; (2) music as a consequence of pre-devised processes; and (3) the notion of "appropriation" as a musical concept. In addition, I will detail the contribution of each topic to earGram's design, in particular how they influenced the articulation between the analysis and composition modules.

5.1 – From Sound to Music: Technical and Conceptual Considerations

5.1.1 - Sampling

In electronic music, sampling (also known as audio collage) is the act of taking a portion of a particular recording and reusing it in a different piece. Apart from previous isolated experiments, musicians began exploring the technique in the late 1940s. The very first sampling experiments were carried almost exclusively in radio broadcast stations, because they had the necessary technology.

The most prominent pioneers of sampling are the French composers Pierre Schaeffer and Pierre Henry; they began to explore experimental radiophonic techniques with the sound technology available in the 1940s at the French Radio in Paris—where the current GRM still resides (Palombini, 1993).

The advent and widespread use of magnetic tape in the early 1950s opened new possibilities to sampling techniques, in particular the exploration of large amounts of audio samples. It is interesting to note that the use of a large corpus of sounds, a crucial feature of earGram, appealed to composers from the very first moment the technology allowed its manipulation. Karlheinz Stockhausen, John Cage, and Iannis Xenakis are three representative composers of the electronic music of this period. Stockhausen used in *Étude des 1000 collants* (1952), known simply as *Étude*, a corpus of millimeter-sized tape

pieces of pre-recorded hammered piano strings, transposed and cropped to their sustained part to assemble a previously devised score that defined a series of pitches, durations, dynamics, and timbres (Manion, 1992). John Cages' *Williams Mix* (1951–1953), a composition for eight magnetic tapes, is another piece from this period that explores the idea of using a large pre-rearranged corpus of sounds as the basis of a composition. *Williams Mix*'s corpus comprised approximately 600 recordings organized in six categories: city sounds, country sounds, electronic sounds, manually produced sounds, wind sounds, and "small" sounds, which need to be amplified (Cage, 1962). Xenakis' compositions *Analogique A* and *B* (1958-1959) and *Bohor* (1962) are also worth mention, not only for its exploration of a large corpus of short sound fragments but also for the assembling process, which was driven by stochastic principles (Di Scipio, 2005).

From the mid 1960s until the 1990s, we witnessed a rapid proliferation of sampling techniques, mainly because of the growing interest of popular music producers and musicians. Sampling featured prominently in renowned bands such as The Beatles, for example in *Tomorrow Never Knows* (1966) and *Revolution 9* (1968), and The Residents, whose song *Swastikas on Parade* (1976) appropriates and samples James Brown extensively.

Later, from the mid 1980s onwards, most electronic dance music has significantly explored samplings techniques. Sampling CDs, a new commercial product that contains rhythmic loops and short bass or melodic phrases, became quite popular among this group of musicians. Commonly, loops featured in these CDs were labeled and distributed by genre, tempo, instrumentation, and mood. Most well known uses of this practice occur in popular music, such as hip-hop, which has immediate roots in the 1960s reggae and dub music of Jamaica, and ancient roots in the oral traditions of Africa.

Sampling techniques have been expanded since the 1940s, importantly including the use of various samples sizes, as explored in micromontage and granular synthesis, two techniques that will be further detailed in the following sections.

5.1.2 – Micromontage

Micromontage defines the process of composing musical works by assembling short audio samples, usually known as microsounds. All sounds between the sample and sound object time scales can be defined as microsounds, roughly equivalent to the range between 10 and 100 milliseconds (Roads, 2001). Micromontage treats sound as streams of acoustic particles in time and frequency domains.

Curtis Roads offers a systematic survey of the history and origins of microsound as well as its application in music composition in his seminal book *Microsound* (2001). Roads not only exposes the history and roots of microsound from the atomistic Greek philosophers of the 5th century BC until the modern concept of sound particles by Einstein and Gabor, but also provides a comprehensive overview of the artistic work done in this domain, including his own compositions.

Iannis Xenakis was the first composer to develop compositional systems that explored microsounds extensively (Roads, 2001)—"grains of sounds" in Xenakis' terminology. For Xenakis, all sounds can be seen as the "integration of grains, of elementary sonic particles, of sonic quanta" (Xenakis, 1971, p. 43). Xenakis developed a taxonomy for grains of sounds and sound-particles assemblages, such as "sound masses," "clouds of sound," and "screens of sound" (Xenakis, 1971).

The Argentinian composer Horacio Vaggione has worked extensively with micromontage techniques and is recognized as a pioneer of using sampling techniques in the digital domain (Sturm, 2006b). Vaggione's first experiments with micromontage date back to 1982 when he started composing *Octuor*. All the sound material used in *Octuor* derives from a set of five audio files that were previously synthesized by the composer. The files were initially segmented into small fragments and later edited and mixed into medium to large-scale structures. *Thema* (1985) for bass saxophone and tape and *Schall* (1995) for tape are two other major works from Vaggione that continue to explore

micromontage. In *Schall*, the composer transforms and arranges thousands of segments of piano sounds to create a variety of textures and themes (Roads, 2001).

The initial experiments of these two composers—Xenakis and Vaggione—constitute the most important impulses in both the theory and practice of micromontage. Their works guided most future developments of the technique, which many composers have continued and extended towards different aesthetic approaches and technology, such as Karlheinz Stockhausen, Gottfried Michael Koenig, and Noah Creshevsky. Of notice the work of the Portuguese composer Carlos Caires, in particular his software IRIN (Caires, 2004), which combines graphic and script editing with algorithmic generation and manipulation of sound sequences to ease the creation of compositions through micromontage. Caires's work points toward interesting directions in regards to how to obtain and compose with very short audio snippets, in particular how to organize and manipulate meso structures.

5.1.3 - Granular Synthesis

Granular synthesis is a technique that assembles very short segments of audio to build sonic textures, which can be understood as an extension of micromontage towards a higher degree of automation (namely in the selection procedures). In fact, a pioneer of granular synthesis—Curtis Roads—was under the supervision of Horacio Vaggione—a micromontage pioneer—while experimenting with the technique. Barry Truax, a Canadian composer and researcher, is another pioneer of granular synthesis in its extension towards real-time uses (Truax, 1988).

Granular synthesis uses short snippets of sound, called grains, to create larger acoustic events. Grains are signals with a Gaussian amplitude envelope that can be constructed from scratch, like different types of sound waves, or short audio segments obtained by segmenting an audio sample. The duration of a grain typically falls into the range of 1-50 milliseconds (Roads, 1998). Most granulators synthesize multiple grains simultaneously at

different density rates, speed, phase, loudness, frequency, and spatial position. Of note is how Barry Truax's soundscape compositions demonstrated that granular synthesis is particularly efficient at generating natural acoustic environmental sounds, such as rain, waterfalls, or animal vocalizations.

Above, I presented an overview of three representative sample-based synthesis techniques in order to introduce the reader to the state-of-the-art technology and artistic practices before the emergence of CSS. Despite having already discussed CSS in several sections of this dissertation,[16] I will address once more this synthesis technique to describe its application in musical composition during the last years.

5.1.4 – Musical Applications of Concatenative Sound Synthesis

In 2006, while referring to Bob Sturm's compositions[17] and to his own compositions using real-time CSS, Diemo Schwarz claimed that "the musical applications of CSS are just starting to become convincing" (Schwarz, 2006, p. 13). Regarding the application of CSS to high-level instrument synthesis, Schwarz (2006) added that "we stand at the same position speech synthesis stood 10 years ago, with yet too small databases, and many open research questions" (p. 13). Schwarz furthers his remarks with a prediction that in a few year's time, CSS will be where speech synthesis is at the time. "After 15 years of research, [concatenative TTS synthesis] now become a technology mature to the extent that all recent commercial speech synthesis systems are concatenative" (Schwarz, 2006, p. 14).

Schwarz's prediction became true regarding the application of CSS to high-level instrument synthesis. The Vienna Symphonic Library,[18] and Synful (Lindemann, 2001) are two remarkable examples of state-of-the-art CSS software for instrumental synthesis.

[16] An overview of the technical components of CSS has been presented in section 1.3 and various aspects of CSS have been discussed in Chapters 3 and 4.
[17] Diemo Schwarz was referring to Bob Sturm's compositions: *Dedication to George Crumb* (2004) and *Gates of Heaven and Hell: Concatenative Variations of a Passage by Mahler* (2005).
[18] http://www.vsl.co.at

Vienna Symphonic Library had several updates for the last years and increased significantly its database.[19] On the contrary, Synful does not rely on its database's size to provide better results, but in additional processing—using transformations of pitch, loudness, and duration. Nonetheless, Synful fulfills the application of high-level instrument synthesis strikingly well.

The application of CSS to instrumental synthesis if of utmost importance for composition, but, even if it improves the quality of the results in comparison to other instrumental synthesis techniques, it does not provide tools that expand a compositional thinking towards new musical ideas. However, these ideas have been explored by different CSS software, such as MATConcat (Sturm, 2004), CataRT (Schwarz, 2006a), and AudioGuide (Hackbarth et al., 2010), and I can summarize them in three major compositional strategies: (1) re-arranging units from the corpus by other rules than the temporal order of their original recordings; (2) composition by navigating through a live- or pre-assembled corpus; and (3) cross-selection and interpolation, which allow to extract and apply the morphology of one corpus to another.

Hitherto the above-mentioned compositional ideas have been mostly applied in musical composition by the CSS systems' developers.[20] A significant exception is Schwarz's CataRT, which has been utilized in many creative projects, even if most of them result from a direct collaboration with Schwarz or from people working at the *Institut de Recherche et Coordination Acoustique/Musique* (IRCAM), where Schwarz currently works. Matthew Burtner, Sebastien Roux, Hector Parra, Luca Francesconi, Stefano Gervasoni, and Dai Fujikura are contemporary music composers that have worked at IRCAM and employed CataRT in their compositions (Schwarz, 2007). Schwarz has also been performing with CataRT for several years, either as a solo performer or in improvisation sessions with live performers. He is a regular presence in the music sessions of many international

[19]The latest Pro Edition Vienna Symphonic Library comprises 235 GB of instrumental sound samples—an increase of 135 GB since its first release in 2002.

[20] Note that my comment may also suffer from a lack of documentation about music composed by CSS. Composers are certainly less concerned with the documentation of the techniques they apply in their practice than researchers that work in the academia.

conferences related to computer music, such as Sound and Music Computing, Live Algorithms, International Computer Music Conference, and New Interfaces for Musical Expression. He has been performing with renowned musicians such as the trombonist George Lewis, the saxophonist Evan Parker, and the clarinetist Etienne Brunet.[21] The last application of CataRT that I would like to highlight is the interactive exploration of sound corpora in combination with new interfaces for music expression. For example, the Plumage project explores sound corpora by navigating in three-dimensional visualizations of the corpora (Schwarz et al., 2007), and the project Dirty Tangible Interfaces (DIRTI) uses CataRT to sonify and interact with tangible interfaces such as granular or liquid material placed in a glass dish (Savary, Schwarz, & Pellerin, 2012).

Norbert Schnell, another IRCAM researcher and head of the IRCAM Real-Time Musical Interactions team, has recently presented the MuBu library for Max/MSP, which is a set of externals for interactive real-time synthesis of analyzed and annotated audio segments. Similarly to CataRT, the MuBu library was already applied as a CSS system in musical composition, notably to assist composers in residence at IRCAM such as Marco Antonio Suárez-Cifuentes in *Caméleon Kaléidoscope* (2010) and *Plis* (2010), and Mari Kimura in *Clone Barcarolle* (2009). MuBu has also been used in projects dealing with new interfaces for music expression such as Mogees,[22] which applies "realtime audio mosaicing" to augment everyday objects and transform them into musical instruments.

Apart from the exception of the work developed at IRCAM and the aforementioned commercial CSS software for instrumental synthesis, Schwarz's prediction about the dissemination of CSS is not yet apparent in contemporary music practice. Most of the remaining compositions or sound examples were mostly produced by the system's developers, as is true of Tristan Jehan, William Brent, Michael Casey, and Ben Hackbarth. I believe that many musicians are interested in the technique, but most CSS software are

[21] The improvisation with George Lewis, Evan Parker and Diemo Schwarz took place during the Live Algorithms for Music conference in 2006, and was later released on CD (Schwarz, 2007). The performance with the clarinetist Etienne Brunet, along with many other examples, is available in Schwarz's website: http://diemo.free.fr.

[22] http://www.brunozamborlin.com/mogees/.

not easy to access, and there are still usability issues that pose major obstacles for its application (some of them have been extensively addressed in this dissertation). In addition, to my knowledge, many CSS systems were never available to the public, and their developers had not provided any sound examples produced by the systems, as the case of Musical Mosaicing and MoSievius.

The popularity of CataRT and more recently MuBu may also be related with the programing environment for which they were developed (i.e., Max/MSP), which is a familiar tool for many artists working in the digital domain. Recently, the technique has also been ported to other programing environments for multimedia production, such as SuperCollider (Stoll, 2011), Pure Data (Brent, 2009; Bernardes, Guedes, & Pennycook, 2013), and ChucK, whose inner structure allows the easy implementation of the technique (Wang, Fiebrink, & Cook, 2007). The dissemination of CSS through various programing environments not only shows an increased interest in the technique during recent years, but also enlarges the possibility for interested users to adopt CSS in the programing environment they are more familiar with.

The recent work of Ben Hackbarth, and in particular his software for CSS named AudioGuide, should not remain unmentioned here due to the weight given by Hackbarth to the exploration of CSS from a compositional and aesthetic standpoint. In his catalogue I would like to highlight Hackbarth's compositions *Am I a Particle or a Wave?* (2011), and *Out Among Sharks, A Moving Target* (2012), which clearly expresses his concept of "sonic transcription" using CSS (Hackbarth et al., 2013).

To conclude, I would like to mention a particular group of systems, whose functionality is highly explored in earGram. I am referring to CSS software that focus on the automatic creation of mashups or stretching an audio source infinitely while retaining its morphology, such as Scrambled Hackz by Sven Koenig, the Plunderphonics' systems presented by Zils and Pachet (2001) and Aucouturier and Pachet (2005), and the Echo Nest Remix API which is an extension of Jehan's (2005) Skeleton. Unfortunately, most of these

systems were never released to the public (e.g., Scrambled Hackz, and the research by Zils, Aucouturier, and Pachet), and only a very limited number of sound examples have been provided, with the exception of Jehan's work, which, is also inaccessible to most musicians, because the Echo Nest Remix API requires advance programming skills to produce musical results. These reasons have motivated me to create a flexible tool like earGram, which is not only adapted to musicians with a traditional educational background, but also freely available on the Internet.[23]

After detailing some of the most prominent sample-based techniques that contributed to the emergence of CSS, along with significant musical applications of CSS, I will focus on the conceptual implications of musical practices that adopt sound as the basis for a composition. I will focus on three main practices: (1) spectral music, as a way of dealing with sound for organizing musical structures; (2) process music, following Steve Reich (1968) terminology to approach composition practices that exclude the need to a detailed low-level specification; and (3) the use of appropriation as a musical concept, which decisively shaped the design of this study, in particular many of earGram's features.

5.1.5 - Sound Structure as a Foundation for Compositional Systems

The use of sound features as a strategy for composing is an attitude not exclusive, but more evident in spectral music. Spectral music was established in the early 1970s by a group of young French composers including Tristan Murail, Gérard Grisey, Hugues Dufourt, Michael Levinas, and Mesias Maiguashca.

In its early days, composers associated with the musical school referred to as spectral music used the analytical possibilities offered by computers to identify, extract, and manipulate sonic properties from audio signals. The resulting analysis allowed the identification of complex patterns, which served as a basis for the extrapolation of

[23] EarGram can be freely downloaded from the following website: https://sites.google.com/site/eargram/download.

musical structures. Spectral music has evolved tremendously since then and today it exhibits a certain flexibility of style that transcends a dogmatic compositional belief system (Gainey, 2009). Since the early adopters of the seventies, many composers, such as Jonathan Harvey and Kaija Saariaho, have adopted, explored, and expanded the scope of action of spectral materials.

These days, spectral music is rather understood as "music in which timbre is an important element of structure or musical language" (Reigle, 2008). In fact, as Grisey notes "spectralism is not a system… like serial music or even tonal music. It's an attitude. It considers sounds, not as dead objects that you can easily and arbitrarily permutate in all directions, but as being like living objects with a birth, lifetime and death" (as cited in Hamilton, 2003). Ultimately, what composers associated with spectral music share is a "belief that music is ultimately sound evolving in time" (Fineberg, 2000, p. 1). Therefore, what is central in the attitude of a spectral composer is the desire to formalize compositional systems based on the structure of sound. Music in this context is rather seen as color, timbres sculpted in time, or a general phenomenon of sound (Fineberg, 2000).

The idea of composing music in which pitch and duration are not the primary elements of musical structure is an important idea behind earGram, which encompasses systematic approaches to explore various dimensions of timbre. However, as Trevor Wishart remarkably articulates, we should be aware that timbre is a "catch-all term for those aspects of sound not included in pitch and duration. Of no value to the sound composer" (Wishart, 1994, p. 135). To overcome the multidimensionality of timbre and allow its use in composition as a musical construct, earGram fragments this sonic attribute in many descriptors. For example, earGram allows the creation of ordered timbres and aggregates of sounds according to sound noisiness—a strategy explored in "spectral compositions" such as Murail's *Désintégrations* (1982) and Saariaho's *Verblendungen* (1984). Another strategy implemented in earGram is the possibility to organize audio units according to psychoacoustic models, namely the use of sensory dissonance, as utilized in the opening

section of Grisey's *Jour, Contre-Jour* (1979).

While addressing the specificity of timbre in sound-based compositions, Trevor Wishart (1994) articulates a related topic of seminal importance here. Wishart suggests that composers should focus on the exploration of the idiosyncratic possibilities offered by the new means of musical production such as synthesizers and the ever-increasing number of human-computer interfaces for music production that are presented every year in conferences such as the International Conference on New Interfaces for Musical Expression (NIME). I pay particular attention to the use of sound parameters dynamically, rather than discrete pitches and fixed durations and dynamics, which are highly attached to the paradigm of traditional Western musical creation for acoustic instruments.

Below, after pointing to the use of spectral music—and particularly the use of timbre—in my current work, I will analyze a strategy that organizes both lower and higher layers of musical structure by (pre-established) musical processes.

5.1.6 - Music as a Process

Another compositional principle that has been extensively explored in earGram is the idea of music as a result of pre-established processes that exclude the need for a detailed note-to-note or sound-to-sound realization. Algorithmic composition falls into this category because algorithms must be expressed as a finite list of well-defined instructions, which in its turn "compose" the score or the aural result of the piece. However, it is not my purpose to discuss in this section the use of algorithms in composition. Instead, I focus on the stylistic features resulting from those approaches.

The term "musical process" is fairly indeterminate in meaning. Erik Christensen (2004) offers a categorization that may help grasp the essence of the term as discussed here, contributing to the term's clarification. Christensen establishes two categories of musical processes: transformative and generative, which, despite the lack of reference, alludes to

Robert Rowe's (1993) taxonomy of interactive music systems' responses.

Transformative musical processes determine all the note-to-note or sound-to-sound details of the composition and the overall form simultaneously by applying various transformations to musical material (Reich, 1968). The performance of musical pieces based on transformative musical processes commonly presents to the listener the genesis of the process.

In this category we find composers such as Steve Reich and Alvin Lucier. Steve Reich's first process compositions were based on tape loops played in two tape recorders out of synchronization with each other, a technique named phase shifting, which produces unforeseen rhythmic patterns. Phase shifting was initially explored in his compositions *It's Gonna Rain* (1965) and *Come Out* (1966). The same technique was later transferred to live instrumental compositions in *Reed Phase* (1966), *Piano Phase (*1967), and *Violin Phase* (1967). Lucier's *I'm Sitting in a Room* (1969) is another example of such an approach. Lucier explores a cyclic repetition of an initially spoken sentence, which is later processed over and over through a recording and diffusion mechanism, altering the nature of the initial signal to the level of rhythmic recognition.

John Cage, Earl Brown, Morton Feldman, and musicians associated with practices such as free improvisation and indeterminacy, work with musical processes that fall into the second category. A clear example of such approach is John Cage's use of the *I Ching*, an ancient Chinese book, in combination with chance operations to devise musical parameters. Cage also used the imperfections in a sheet of paper to determine elements of a musical score, namely pitches. This last technique was greatly explored in *Music for Piano* (1952–1962).

A seminal distinction between the musical pieces of this group in relation to those of the first is that the compositional processes cannot be heard when the piece is performed—the musical processes and the sounding music have no audible connection. Also, contrary to the first approach, which eliminates any possibility of improvisation, this

category extends the role of the interpreter to a high degree of interference in the creative process, because many elements of musical structure remain undetermined.

The musical processes explored in this study are mainly, but not exclusively, located in the second category, that is, generative and rule-based. However, with very few adjustments, the system may be adapted to incorporate other techniques, including transformative ones. The raw material that is manipulated relies on existing audio sources. Therefore, the final aspect that I would like to focus on is the aesthetic implications of using pre-recorded material in the compositional design, particularly the appropriation of musical works.

5.1.7 - Appropriation as a Musical Concept

The use of existing musical material as a basis for a new composition dates back to ancient times and cannot be fully detailed here because it goes beyond the core subject of this dissertation. However, I will provide a general overview of the subject because the aesthetics behind this attitude are present in earGram's compositional design. The appropriation of musical material as a composition prerogative is as old as polyphony. The practice was mainly explored in two different ways: (1) by composers that refer to their previous works, and (2) composers that base parts of their works on material from others. When composers integrate material from others' music in their compositions, they usually refer to contemporaries affiliated stylistically (Griffiths, 1981).

Between the 12th and 15th centuries, composers frequently used pre-existent melodies as a base for new compositions, particularly in motets. These melodies, named Cantus Firmus, were usually taken from Gregorian chants, and generally presented in long notes against a more quickly moving texture (Burkholder, 1983). Another significant example of music appropriation occurs between 17th and 18th centuries amongst the numerous

composers of Bach's family legacy (Geiringer, 1950).[24]

Until the 20th century, composers who integrated pre-existing music into their pieces adapted the material to their idiom, and their compositions maintained a sense of stylistic unity. Contrarily, appropriation in the 20th century shifted towards the use of "ready-made" musical material that "clashes with the prevailing style of the original piece, rather than conforming to it" (Leung, 2008). The neoclassical works of Igor Stravinsky, such as *Pulcinella* (1920) and *The Fairy's Kiss* (1928), are remarkable examples of compositions in which Stravinsky reworked upon a borrowed material. Stravinsky does not appropriate for increasing his own expressivity, but rather for expressing his view of the past (Leung, 2008).

The idea of "ready-made" or collage is even more present in the works of Charles Ives and George Crumb. In *Central Park in the Dark* (1906) and *The Fourth of July*, the third movement of *A Symphony: New England Holidays* (1897-1913), Ives presents to the listener a complex interaction between his "imaginary present" and "memorable past." Ives commonly refers to the past by quoting his childhood tunes (Leung, 2008). Crumb appropriates musical material from others by literally quoting the material in his compositions. In Crumb's compositions appropriated musical materials cohabit independently, integrating and overlaying uneven aesthetics. A remarkable example of Crumb's use of appropriation can be found in *Night Spell I*, the sixth piece in *Makrokosmos* (1972-1973).[25]

Another notable example of music appropriation in the 20th century, which cannot remain unmentioned is the third movement of Berio's *Symphony* (1969) for eight singers and orchestra, which was entirely conceived as a tapestry of quotes from various works by the following composers: Bach, Beethoven, Brahms, Mahler, Debussy, Ravel, Strauss, Stravinsky, Schoenberg, Berg, Stockhausen, Boulez, and even early works by Berio himself

[24] Please refer to Burkholder (1983) and Leung (2008) for a comprehensive review of appropriation techniques in early Western music.
[25] For a deeper review on appropriation techniques used by 20th century composers please refer to J. Peter Burkholder (1983, 1994), who systematically outlines a large set of "borrowing" techniques found in music with a particular emphasis on the musical pieces of Charles Ives.

(Altmann, 1977).

From the 1940s onwards, the practice of appropriation became popular due to technological advances that allowed musicians to record, manipulate and playback audio by electronic means. The gradual massification of music technology tools—in particular the sampler—since the 1940s, provoked an aesthetic shift from an early historical phase designated as acousmatic to a later stage addressed commonly as sampling culture (Waters, 2000). While the first relies mostly in self-referential matter and on the listening experience, the second relies on musical and cultural referential contexts, notably by incorporating and reutilizing pre-existing music recordings data to convey new means of expression (Waters, 2000). As I mentioned earlier, the sampling technique relies on existing recordings and is therefore related to the concept of appropriation as a compositional principle. In fact, it is only in the second half of the 20th century that the term appropriation became a musical concept (Landy, 2007).

The first example of an electronic music composition entirely based of borrowed audio material is James Tenney's 1961 composition *Collage #1 (Blue Suede)* (Cutler, 2004). In this composition, Tenney recombines and manipulates sound material from Elvis Presley's song *Blue Suede Shoes*. Two additional early examples of compositions that explicitly expose the technique of appropriation are Bernard Parmegiani's *Pop'eclectic* (1968) and *Du pop à l'âne* (1969). These tracks were created as tapestries of mostly late 1960s pop records, and assembled with unique and significant relationships between sonorities, genres, and cultural contexts by transitioning seamlessly between small samples. Tenney's and Parmegiani's works also question the distinction between low art and high art sometimes also referred to as popular music and art music. Since then the differences between these categories have become less prominent (Emmerson, 2001; Landy, 2007). Another proponent of appropriation in electronic music who explores this overlap between low art and high art is John Oswald. His 1988 CD named *Plunderphonic* (Oswald, 2001), demonstrates an unusually broad eclecticism by plundering, recombining, and

decontextualizing music from Ludwig van Beethoven to the Beatles.[26]

The practice of appropriation is even more evident in popular music, namely after the emergence of affordable technology such as the sampler, which was and still is a huge catalyst of the technique. Many concepts are associated with appropriation and expose similar or overlapping approaches, such as sampling, remix, collage, mashup, cutups, cut & paste, blend, crossover, plunderphonics, etc. All of these terms are highly associated with popular music, and in particular with practices and styles such as Hip-hop, Rap, and DJing.

The idea of appropriation has been explored in many other fields, which to a certain extent have also influenced many contemporary composers. The idea of appropriation is particularly present in the visual arts. The collages of George Braque and Pablo Picasso, and the ready-mades from the artists associated with the Dada movement are clear examples of such. In literature, an exponent of the cut-up technique, that is, a literary technique in which a text is cut up and rearranged to create a new text is the American writer William Burroughs. In philosophy, I may cite Mikhail Bakhtin, in particular his concept of dialogisms, which has been acknowledge and followed by Julia Kristeva in her intertextual theory (Kristeva, 1969).

The system developed here embraces the idea of appropriation by recombining user-assigned sounds. In comparison with most CSS systems, earGram uses relatively larger sound segments, whose source is easily recognizable after recombination. Therefore, the resulting music can be seen to a certain extent as a remix or variation of the audio source(s). In addition, if one uses a corpus that comprises sound objects from audio sources with distinct styles, origins, or aesthetics, one may not only recombine sound objects according to morphological features, but also drawing upon the cultural associations of the original pieces.

A final note should be paid to the relation between copyright laws and the practice of

[26] Please refer to Oswald (1986), Holm-Hudson (1997), and Cutler (2004) for an historical and conceptual overview of Oswald's work.

appropriation. As Simon Waters (2000) points, sampling embeds an ambiguous relation between ownership and authorship. The practice of appropriation raises many problems concerning copyright infringements. I will not unpack the topic here, because it is not of primary importance to my dissertation. However, the reader may refer to Bob Sturm (2006a) for a legal discussion on the subject within the scope of sound synthesis, and Lawrence Lessig (2008, 2004, 2001) for a general take on the subject.

Having situated earGram historically and aesthetically, I will narrow my perspective to the practical implications of the various technical and conceptual issues raised in this chapter. In order to do so, I will first discuss design strategies for musical composition (§ 5.2), which will then be examined from an algorithmic perspective (§ 5.3) and more precisely in the devised framework (§ 5.4).

5.2 – Design Strategies for Musical Composition

As Gottfried Koenig (1978) points out, it is interesting to note that the concept of musical composition relates to both the act of producing a score or a fixed media work, and to the result of that process. While the concept can be seen as definite in terms of the resulting product, it says nothing with regard to the creative process. It is important to understand the creative process, however, in order to be able to encode it algorithmically (or at least partially) and ultimately generate some coherent musical results.

A crucial feature of any computational system that intends to automate the processes of music creation is the need to algorithmically encode some creative features that are inherent to human activity. However, creativity is an extremely difficult concept to circumscribe in a strict definition, in particular because there is a lack of understanding of how our creative mechanisms fully work (Taylor, 1988; Csikszentmihalyi, 2009). Even though computer programs seem to oppose to the idea of limitless originality, they also

offer potentialities that are hardly achieved by humans. Computers can actively contribute to the development of new creative practices and promote interesting discussions concerning artificial creativity.

Composing can be seen as a decision-making process. Many choices have to be made during the creation of a musical piece, from high-level attributes such as instrumentation to low-level elements, such as pitches and durations. Musical composition design and practice commonly require one of three distinct approaches: (1) top-down, (2) bottom-up, or (3) the combination of both (Roads, 2001).

A top-down approach to musical composition starts by developing and formulating the macrostructure of the work as a predetermined plan or template, whose details or lower-level formulation are elaborated at later stages of the composition process. All time scales below the macrostructure are considered and refined in greater detail according to the initial plan, until the most basic elements of the structure are elaborated. In Western music, this compositional strategy has been extensively adopted from the 17th to the late 19th centuries, especially because during this historical period the form or macrostructure of the works was mainly confined to a limited number of options (Apel, 1972), such as the sonata form, the rondo, and the fugue. Many music theory textbooks catalog the generic classical forms (Bennett, 1981), whose widespread use enters into a decadent phase at the turn of the 20th century.

By contrast, a bottom-up approach conceives the musical form as the result of a process. The macrostructure is the consequence of the development of small-scale ideas or provoked by the interaction of the lower levels of musical structure. Roads (2001) mentions serialism as a paradigmatic example of a bottom-up musical compositional technique, in particular the permutations resulting from applying the inversion or retrograde operations. Bottom-up compositional strategies may also be found in electronic music in processes such as time-expanding a sound fragment into evolving “sound masses.” These examples create an apparent line between different historical periods.

Top-down approaches were assigned to musical compositional practices before the 20th century and bottom-up strategies from the 20th century onwards. Although some generalization may be made in this regard, the implied distinction is not entirely true. Not only has the musical form evolved continuously from the 17th to the 19th centuries, but also in contemporary music the older forms remain present. This is not to say that the use of preconceived forms has died. The practice of top-down strategies in contemporary music still subsists (Roads, 2001), even if in most cases it does not apply to known forms.

The compositional process may also incorporate both top-down and bottom-up approaches. In the case of what I call the hybrid approach, the composition is the result of a constant negotiation between its low- and high-hierarchical layers, which are drawn simultaneously.

In electronic music the creative process is not dissimilar from traditional instrumental composition concerning the various levels of decision-making. However, it is possible to point to a clear difference between the two practices, which are related to the nature and idiosyncrasies of the raw material used. While instrumental music departs from an abstract to a concrete realization, electronic music commonly starts from concrete sounds (or synthesis methods) to an abstract level. Therefore, in electronic music the act of composing involves the need to define the elementary units of the composition, that is, the sounds themselves. As Koenig notes "electronic sounds or graphic symbols are not always additions to composition; they are often 'composed' themselves, i.e., put together according to aspects which are valid for actual composing" (Koenig, 1978). Koenig's statement articulates a fundamental aspect of electronic music compositional processes, which must also be taken into consideration in CSS during the choice of the audio source(s) and segmentation strategies. The synthesis quality of earGram is not only dependent on the characteristics of its database, but also on the algorithmic composition strategies for unit selection.

5.3 - Algorithmic Composition

Algorithmic composition is the term used to describe “a sequence (set) of rules (instructions, operations) for solving (accomplishing) a [particular] problem (task) [in a finite number of steps] of combining musical parts (things, elements) into a whole (composition)” (Cope, 1993). David Cope’s definition of algorithmic composition is one of the broadest and most concise descriptions of the field, especially because it does not imply any means of production. It not only encompasses the various historic periods that presented work in this domain, but also restricts its *modus operandi* to a set of specific and clear procedures. The definition comprises two parts. The first part addresses the general definition of algorithm (Knuth, 1968; Stone, 1972) and the second part restricts the target object of the algorithm problem-solving strategy to the music domain. For Cope, music is defined as an activity that groups musical elements into a whole composition.

When designing an algorithmic work, the role of the composer is significantly different from the attitude undertaken in traditional Western compositional approaches. Heinrich Taube (2004) refers to this role as a new compositional paradigm. While creating an algorithmic composition, the composer works on a meta-level because instead of outlining a piece by stating musical events notation or soundwise, he/she designs a model which in turn generates the work.

An algorithm, within this domain, constitutes a well-defined set of instructions that define and control particular aspects of the composition. The algorithm must effectively provide a finite number of states and their interaction. However, it does not necessarily convey deterministic results. Algorithms for music composition are commonly initialized by data that alters its behavior, and consequently its outcome.

5.3.1 – Algorithmic Composition Approaches

The earliest experiments on algorithmic composition are commonly traced back to the 11th century (Roads, 1996). Apart from their historic importance, algorithmic music composed before the mid 20th century constitutes isolated experiments with minor significance to the music field. Algorithmic composition establishes itself as a field in its own right in the late 1950s by integrating the power of digital computers in the design of algorithms to assist the generation of musical works.[27]

Early approaches to CAAC—by Lejaren Hiller, Leonard Isaacson, Isacson, Iannis Xenakis, and Gottfried Koenig—have established the basis of the practice according to the following two major approaches: (1) generative models of music for style imitation and (2) generative models of music for genuine composition.[28] Several composers and researchers have further research in CAAC according to these two major trends.

The first line of research, whereby generative models of music for style imitation, follows the early experiments of Hiller, Isacson, and Koenig—in particular the formalization of principles from music theory or the emulation of a particular style, composer, or body of works. There are two approaches to generative models for style imitation: (1) knowledge engineering, in which the generation is guided by rules and constraints encoded in some logic or grammar, and (2) empirical induction, in which the generation relies on statistical models resultant from the analysis of existing compositions (Conklin & Witten, 1995).

Some of the topics that have been continuously revisited within the knowledge engineering approach to generative models of music for style imitation are: the generation of species counterpoint (Ovans & Davidson, 1992; Farbood & Schoner, 2001); functional

[27] For a comprehensive review of the history of CAAC please refer to Nierhaus (2009) and Ariza (2005).
[28] The concept of genuine compositions establishes a distinction between approaches to music composition, whose starting point for a work focus rather on idiomatic approaches that a clear desire to imitate the style of a particular composer, work(s), period, etc. The concept does not intend to raise aesthetic questions related to the originality and/or validity of a work, or even its definition as art. I am only concerned with distinguishing an attitude towards the act of composing.

harmony as used in Western music from the 17th to 19th centuries (Pachet & Roy, 2001); the automatic generation of rhythmic events, namely in the context of interactive music systems (Eigenfeldt, 2009; Bernardes, Guedes, & Pennycook, 2010; Sioros & Guedes, 2011); and the exploration of serial music operations (Essl, 1995; Ariza, 2004). Concerning the empirical induction methods to generative models of music for style imitation one may highlight the work of David Cope (1993, 1996, 2001). Cope (1996) extensively used transition networks to create representations of musical data extracted from one or more compositions. These representations allow the automatic generation of new pieces that resemble the style of analyzed compositions.

The second line of research in algorithmic composition—genuine compositions— applies techniques that are commonly inspired by models outside of music, or formulated from scratch. This research line is grounded in the early algorithmic music experiments carried out by Xenakis, Hiller, and Isacson. The adoption of terminology, concepts, and algorithms from disciplines outside the music domain, in particular from biology, became popular in this domain from the 1980s onwards. Typical examples of such strategies are cellular automata (Beyls, 1989; Miranda, 2001), chaos attractors (Pressing, 1988; Bidlack, 1992; Leach & Fitch, 1995), Lindenmayer systems (Prusinkiewicz, 1986; DuBois, 2003), and artificial neural networks (Hild, Feulner, & Menzel, 1992; Mozer, 1994).

Current commercial digital audio workstations, such as (Magic) Garageband[29], Ableton Live[30], and in particular the programming environments for interactive music creation, such as Open Music,[31] Max/MSP,[32] Pure Data,[33] and PWGL[34] provide a large set of tools for the exploration of CAAC techniques.

[29] http://www.apple.com/ilife/garageband/.
[30] https://www.ableton.com/en/live/.
[31] http://repmus.ircam.fr/openmusic/home.
[32] http://cycling74.com/products/max/.
[33] http://puredata.info/.
[34] http://www2.siba.fi/PWGL/.

5.4 – Computational Life Cycle of Music: An Analysis-Synthesis Approach

In this section, I establish a link between the previous sections of this chapter, which provide theoretical and practical foundations of composition, and the generative music strategies developed to recombine sound objects. I will start by describing the architecture of the framework developed here, and conclude by detailing the interaction between the two major modules of the framework (i.e., analysis and composition) and how they determine compositional strategies.

The proposed framework can be summarized and characterized by a compound word: analysis-synthesis. Analysis-synthesis defines two complementary procedures that were addressed in two distinct parts of this dissertation. The first is rather analytical, and its aim is threefold: (1) decomposing the audio continuum into elementary units, (2) describing the content of the audio units, and (3) modeling and depicting the higher structural levels of the audio source(s). The counterparts of these operations are the generative aspects of the framework. Both modules examine the same time scales, but assume an inverse or complementary path. While analysis deconstructs the audio continuum from lower to higher elements of musical structure, the composition module recombines the units using algorithmic strategies, from the higher to the lower levels of musical structure.

The proposed framework can be placed along the axis between automatic music generation and assisted-algorithmic composition, merged in the concept of CAAC. Additionally, the generative strategies implemented in the framework are not exclusively focused on style imitation or genuine algorithmic strategies, but rather on a hybrid combination of these. As a matter of fact, in addition to a consistent description of all audio units, the analytical module goes as far as modeling individual elements of the audio source(s) structure, which can be used to automate parameters of the generated music.

Therefore, earGram does not create a comprehensive representation of the audio source(s) entire dimension or their inter-relationships.

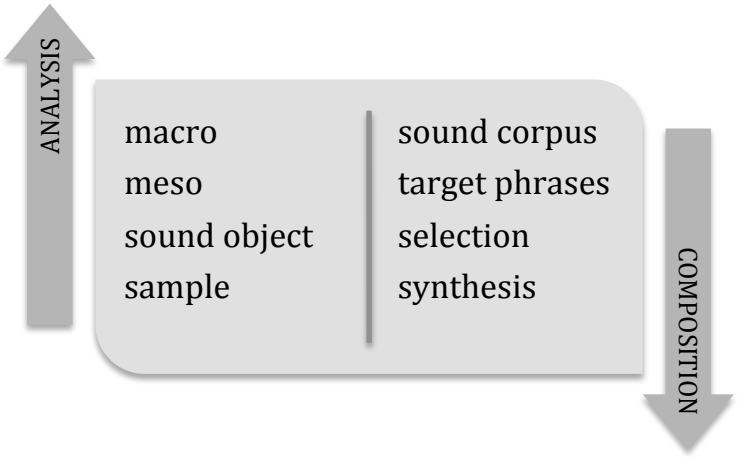


Figure 5.1 – Relationship between the components of the analysis and composition modules of the framework.

The diagram depicted in Figure 5.1 not only exposes and summarizes the complementarity of the analysis and generation blocks, but also reveals another particularity that was planned while designing the framework. It is possible to draw an inverse relation between all constituent blocks that compose the two major modules of the framework. Even if the two modules have quite distinct goals, they have a unifying feature: both organize its various operations according to hierarchical layers of musical structure. The links are established between the various analyzed time scales and the hierarchical levels of the composition process. For each analyzed time scale there is a correspondent task that imposes constraints in the selection process of the generative algorithms. Parallels can be established between the time scales of the analytical module and the hierarchical structure of the composition module (see Figure 5.1).

The following four parallels may be established between the two modules (with the path of the composition module as a primary reference): (1) definition of a sound corpus, or various sub-spaces of the corpus to the macro-level analysis provided by the

visualization strategies, clustering algorithms, or key-induction algorithm; (2) the definition of target phrases has its parallel with the meso structure of the audio source(s); (3) the algorithmic selection procedures work at the same level as the description scheme devised to characterize the units of the corpus and the similarity computation measures; and (4) synthesis, the last module of the composition chain, has a parallel with the sample time scale, because both constitute the most extreme elements of musical structure considered here and correspond to the departing and final matter of the system. All these relationships will be better understood when examined in a practical application in the next chapter.

Chapter 6

Content-Based Algorithmic-Assisted Audio Composition

In the current chapter, I detail the strategies for generating music in earGram. I start by focusing on methods for "composing" a corpus of units and their organization into larger sections (§ 6.1). Next, I focus on the low-level aspects of the generation and describe the following four generative music strategies (also addressed as playing modes or recombination methods) that function as unit selection algorithms in earGram: (1) spaceMap, (2) soundscapeMap, (3) shuffMeter, and (4) infiniteMode. I organized the four playing modes into three sections corresponding to the types of generative strategies applied: (1) micro-time sound design (§ 6.2); and generative models for style imitation based on, (2) knowledge engineering (§ 6.3), and (3) empirical induction (§ 6.4).

The chapter ends by detailing the methods implemented in earGram for synthesizing sequences of audio units selected by the generative methods (§ 6.5), and describing

musical compositions created almost exclusively by earGram (§ 6.6). For each generative strategy, I created several sound examples that not only demonstrate individual features of the software, but also the full creative potential of each of the generative strategies. The numbered sound examples referred throughout the chapter can be found in the accompanying CD. The reader is also invited to explore earGram—provided in the accompanying CD—in order to understand the musical results produced by each of the generative strategies.

6.1 – Composing a Corpus of Sound Units

The need to create the elementary units of the composition in electronic music practices is of utmost importance in the compositional process, and may be seen as similar to the choice of instrumentation in acoustic instrumental compositional approaches.

In earGram, the elementary (audio) units that compose the corpus result from a rather automatic operation. However, the user is not only responsible for choosing the audio source(s) from which the units are devised, but also for setting important parameters that guide the type of units adopted. These two elements have tremendous implications in the synthesis quality and should be defined according to the target characteristics and/or the application context. The audio sources constitute the raw audio data that is concatenated at the final stage of the algorithm, and the segmentation strategy imposes severe constraints in the creative and technical possibilities of the algorithm, such as the balance between the identification of audio source and the target phrase, and the computational costs associated with the use of shorter audio units. In sum, the user must consider three fundamental properties while creating the corpus: (1) the characteristics of the audio source(s), (2) the application context, and (3) the duration of the units.

Despite earGram's design as an "agnostic" music system that processes any type of digitized music, regardless of genre or style, the user must be aware that in order to

synthesize a soundscape composed of environmental sound sources, one should not rely on collections of units segmented on a beat basis with a strong sense of pulse. To impose limits or define fixed solutions in creative contexts may be an inconsiderate option; nonetheless, for the sake of clarity, I will pinpoint certain directions and association between application contexts and preferable segmentation strategies.

A possible distinction can be established between playing modes that are meant to work with units that were segmented according to a found pulse and more irregular non-uniform units. The playing modes spaceMap and soundscapeMap can be assigned to the last category, the playing mode shuffMeter to the first category, and finally the playing mode infiniteMode to both categories. In sum, the two types of units can be associated with the following two application domains: (1) (polyphonic) music with a strong sense of pulse and (2) soundscapes.

Unit duration is another parameter in CSS that has tremendous implications in the quality of the synthesis, in terms of regulating the identification of the source(s) in the synthesized signal. Concisely, unit duration is proportionally inverse to the identification of the audio source(s). While units with longer duration tend to preserve the acoustic identity of the source(s), units with smaller duration tend to emphasize the identity of the target. In earGram, the units' duration is confined to sound objects, which offer a strong identity of the audio source(s) in the synthesized output of the system. This particular feature is of utmost importance in earGram because the target phrases are commonly defined according to features extracted from the audio source(s). Therefore, the system's output focuses to a large degree in the generation of audio source "variations," by preserving the morphology of the source and its sonic identity.

The audio units' duration can also be related to the quality of the units' descriptions. The descriptions of shorter units also tend to be more accurate in comparison with longer units. The descriptions of longer units may sometimes provide crude, erroneous, or inaccurate descriptions because they often reduce successive analysis windows to single

values. However, the adoption of longer units tends to synthesize signals with greater naturalness, because there are less concatenation points. In earGram, the accuracy of the units' descriptions reduce significantly the audio signal data to a minimal representation, but still provide valuable information for the musicians who want to manipulate them, as verified in the generative musical strategies present in earGram and detailed in the next sections. Concerning the concatenation points, the segmentation strategies adopted in earGram—onset detection and beat tracking—provide a greater degree of naturalness in the recombination/concatenation phase of the system.

Two other technical implications should be considered when defining a segmentation strategy, which may impose changes in the units' duration: (1) the computational cost involved, and (2) the amount of memory required. EarGram has reduced computational costs due to the relatively large duration of the audio units on both the analysis and selection phases of the system, allowing faster matches. The preferred use of sound objects over audio units with smaller duration also requires less memory space to store the units' content analysis.

A final note should be made in regards to the descriptions used to characterize audio units and its suitability for two application contexts described. In order to describe the audio units' content, earGram adopts the description scheme detailed earlier (§ 3.4). The use of musicological studies grounded in the principle of reduced listening to devise a description scheme allows the characterization of the totality of sounds perceivable by humans disregarding its causes, genre, stylistic idiosyncrasies, or means of production. The descriptors only reveal abstract characteristics of sound objects and are consistent for describing any audio units within the sound object time scale. The resultant descriptions also allow the comparison between non-uniform audio units with consistency. Therefore, one needs only to focus on the suitability of the audio source(s) and their segmentation strategy to the application context—disregarding the choice of appropriate descriptions adapted to the nature of the audio source(s).

6.1.1 – Planning the Macrostructure

I designed earGram with an awareness of some CAAC limitations, particularly the difficulty of organizing the higher layers of musical structure. Style imitation algorithms, for instance, tend to be very efficient in the generation of musical results that resemble the original data on a moment-to-moment basis. However, the output of such stochastic processes tends to fail at emulating the higher layers of musical structure (Cambouropoulos, 1994; Jacob, 1996). In this section, I propose an intermediary solution to generate longer pieces with consistency that minimizes the drawbacks of applying CAAC by providing some tools that may guide the user.

In order to provide some control over the higher layers of musical structure, I adopt a technique that resembles the type of low-level decisions in CSS. The strategy is based on selection procedures. The user has the possibility to constrain the corpus to sub-spaces that can be easily chosen and interchanged during performance. Therefore, the user has not only real-time control of the target definition, but also of the sub-spaces defined in advance.

The sub-spaces in earGram need to be manually assigned prior to performance. However, the system provides useful information, notably visualizations of the corpus that can guide the definition of corpus sub-spaces. An example of such is the possibility to interactively draw sub-spaces on top of the self-similarity matrix visualization (see Figure 6.1). In addition, the decision process can become virtually automatic when combining visualization strategies with the implemented clustering algorithms. In that case, the user merely needs to map the automatically created clusters to sub-spaces.

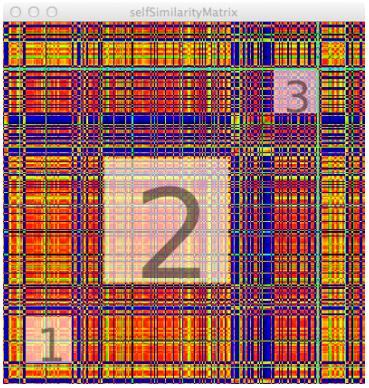


Figure 6.1 - Three sub-spaces of the corpus defined on top of self-similarity matrix visualization.

A more refined strategy for constraining the corpus is to assign limits to the set of available descriptors in the corpus (the constraints can be specified in the interface shown in Figure 6.2). By reducing the descriptors' range, some units may be excluded from the corpus. An interesting use of this tool is to regulate parameters that are not considered in the generation of new sequences. For example, if the target phrase of the generative strategy considers only the units' loudness, it is possible to regulate the use of very noisy sounds by constricting the noisiness descriptor to a narrow band. It is important to note that the results differ when the two descriptors are specified in the target.

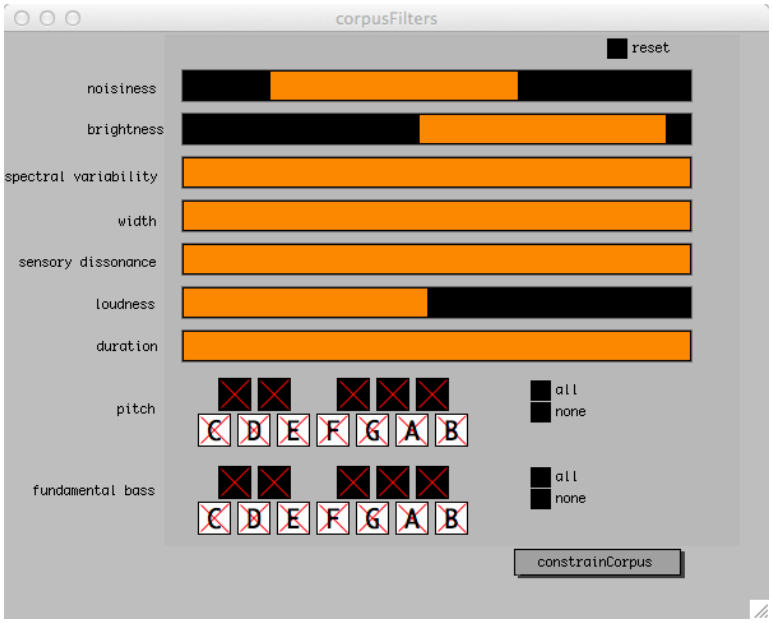


Figure 6.2 - Software interface that allows a user to constrain the corpus in earGram.

The presented solutions for organizing the higher structural layers of the generated music result from my motivation to design software with which I am able to play in musical contexts outside of the lab, but do not solve the essence of the problem. It is also important to bear in mind that in order to create relevant artistic results more research in CAAC is necessary (Jacob, 1996), namely to solve the problem of the organization of the meso and macro structures of generated music.

As David Cope (1984) notes, Hiller's early CAAC experiments lack artistic success. However, these early experiments in algorithmic composition and their later impact in the work of many researchers/composers significantly increased our conception of music, namely by allowing us to test different compositional theories and redesign their *modus operandi*. Therefore, these experiments are intellectually stimulating, and time will tell whether or not they established the ground for the development of relevant artistic output.

After addressing the organization of the higher structural layers of the generated music in earGram, I will detail four unit selection algorithms—spaceMap, soundscapeMap (§ 6.2); shuffMeter (§ 6.3); and infiniteMode (§ 6.4)—that automatically generate unit sequences (low-level elements of musical structure) according to pre-defined algorithms.

6.2 - Micro-Time Sonic Design: SpaceMap and SoundscapeMap

The first two playing modes, spaceMap and soundscapeMap, are addressed together because they share many features. The most distinctive among these common features are how they synthesize sound masses with variable density, and how they use a navigable graphical interface to define targets. This last feature is expressed in the names of the modes by the suffix "–Map."

Both strategies can be seen as extended granular synthesizers with an extra layer of control over the acoustic results. This refined control over the synthesis is driven by the

organized representation of the units in the descriptor space, which allows the creation of sonic textures with highly controllable nuances. Although the idea of variable densities may be contrary to the very concept of concatenation, simultaneous events have been extensively explored in CSS literature (Hackbarth et al., 2010, 2013; Schwarz, 2012). The outcome of these two modes can reach a dense "cloud of sounds" as a result of an arbitrary number of overlapping units and explore processes such as coalescence (cloud formation), and evaporation (cloud disintegration) in sonic form.

The title of this section is partially borrowed from Agostino Di Scipio's 2009 article in the *Contemporary Music Review*, and denotes an application context and a preferable music time scale that I consider in relation to these two playing modes. In "Micro-Time Sonic Design and Timbre Formation", Di Scipio (2009) outlines a particular approach to composition, in which the formation of musical structures emerges from the assemblage of micro-level units driven by microstructural processes. I direct particular focus to the notion of timbre as the primary element of musical structure, which fosters a transformation of compositional paradigm (Di Scipio, 2009).

SpaceMap and soundscapeMap are particularly adapted to musical practices or applications such as sound design for movies, video games, or installations, because they provide a useful set of tools to explore soundscapes whose behavior may be dynamically changed according to input data.

6.2.1 - SpaceMap

SpaceMap synthesizes spatial trajectories defined by navigating in a 2D-plot representation of the corpus (see Figure 4.3). Besides the creative potential of this target definition, spaceMap also allows an intuitive exploration of the corpus, which may be of primary importance when dealing with unknown audio sources. The technique has been greatly explored in CSS, in particular by Diemo Schwarz (Schwarz et al., 2008; Schwarz,

2012). I extend similar approaches to spaceMap in two aspects. The first aspect is related to the strategies used for plotting a corpus of sound units adapted to musical composition imperatives. The second aspect is the incorporation of tendency masks, a CAAC strategy, in the algorithmic chain. While the first aspect has already been addressed, the adoption of tendency masks exposes a major concern of my research: how to extend known CAAC strategies towards the adoption of audio-content descriptions, and will be extensively detailed in this section. I will mainly focus on how tendency masks may help enhance expressive phrasing, mostly by allowing a larger degree of freedom in the target definition.

The interface of spaceMap is a two-dimensional plane whose axes may be assigned to any of the descriptors (or any combination of them) from the description scheme. This representation corresponds to a visualization strategy named sound-space (§ 4.4.1). The visual representation of the corpus is the point of departure for defining targets in the descriptor space as spatial trajectories. While small movements synthesize similar-sounding units, larger movements pick grains with greater sonic differences. The feature space is easily changed during performance by switching between four available presets.

SpaceMap encompasses the following four modes of interaction to define spatial trajectories or targets, referred to in the software as trigger modes: (1) *continuousPointer*: continuously plays units at a user-defined rate according to the controller's position on the interface; (2) *pointerClick*: follows the same method as point 1, but the units are played in response to a controller command; (3) *colorPicker*: selects units based on RGB color values retrieved from a navigable color grid; and (4) *liveInput*: plays units at a user-defined rate and maps the pointer position to coordinates provided by the analysis of a live input source. Various controllers may steer the navigation and adopt most of the aforementioned trigger modes. The software implementation only adopts the mouse and audio signals as input controllers; however, with very few adjustments to the code, it is easy to implement any other controller with similar or higher degrees of

freedom.

The target specifications of spaceMap are the coordinates of two-dimensional points, which represent the controller's trajectory. For each of the target's point, the unit selection algorithm retrieves its closest unit. A random degree of variability may be assigned to the target, which allows the creation of rich textures by slightly deviating around a point's coordinates and also avoids the continuous repetition of the same units or the creation of undesired rhythmic patterns if the controller is stationary. The user may also specify a random degree of variability in the playing rate of the units of trigger modes 1, 3, and 4.

SpaceMap also allows the control of three other parameters in real-time: (1) amplitude, (2) pitch, and (3) spatial location. Similarly to the target specification and the playing rate, all aforementioned parameters may also adopt random degrees of variability. Despite the refined control the parameters offer over the synthesis, their manipulation during performance is highly limited because the user is already busy defining trajectories in the interface. To overcome this limitation the software allows the definition of automations for any of these parameters, which outline its evolution in time with fixed values or through the use of tendency masks.

From a creative point of view, the implementation of random deviations in some parameters allows the creation of sonic variations of the same controller's gesture, which can also be seen as a variation of a musical phrase. Sound example 1 demonstrates this strategy by synthesizing the same trajectory three times using the following pairs of features: (1) fundamental bass and spectral variability; (2) width and sensory dissonance; and (3) noisiness and loudness.

Tendency masks are well-known strategies for the generation of musical structures that can be understood by their graphical representation. The use of tendency masks allows the user to define the direction of a parameters' gesture, which is decided stochastically according to an assigned range. To regulate tendency masks the user defines

upper and lower limits at particular times. The tendency masks implemented in earGram allow the control of density, degree of random deviation from the pointer position, gain, pitch, and spatial position. The tendency masks can be specified in earGram by enlarging its range and changing the direction of the curve on the interface (shown in Figure 6.3). Each pixel corresponds to a user-assigned duration. Therefore, one can experiment with the same curves but in different time frames. In other words, the representation may be stretched while retaining the same characteristics.

Tendency masks have been used in earGram to demonstrate how algorithmic strategies may assist the definition of targets in spaceMap. Many other CAACs, in particular methods commonly used in the manipulation of symbolic representations of music (such as cellular automata, fractals, and Lindenmayer systems) can be implemented in the framework for the same purpose as tendency masks.

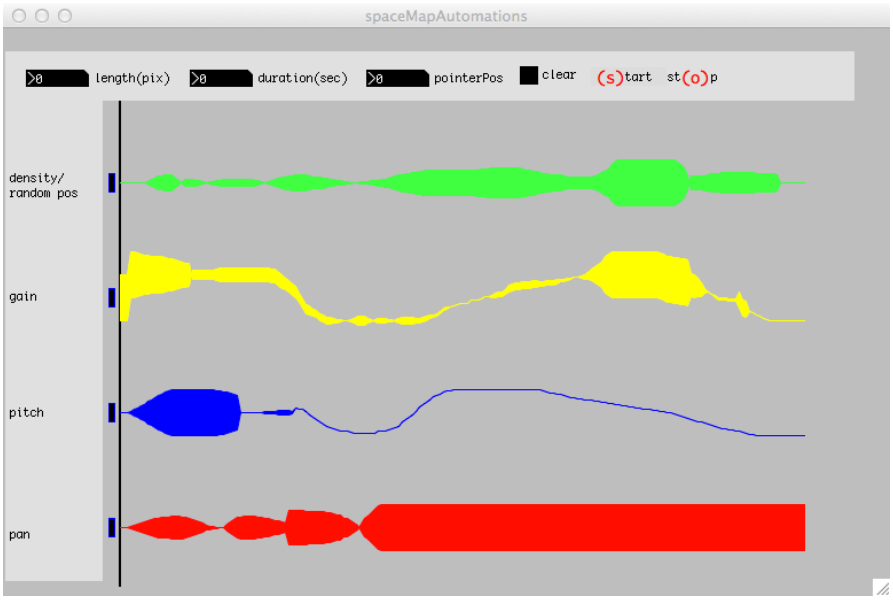


Figure 6.3 - Software interface that allows a user to specify the automation of several parameters of the playing mode spaceMap in earGram through the use of tendency masks.

The software also allows the creation of several bus-channels that support the incorporation of audio effects or combinations of them (e.g. spectral morphing, flanger,

chorus, reverberation, filters, etc.). The units may be separately routed to different bus-channels created in advance.

6.2.2 – Playing with a Live Input Audio Signal

SpaceMap is the only playing mode in earGram that can interact in real-time with a live input audio source. EarGram can process live audio signals in two different ways: (1) as a target (by translating the audio representation in a collection of audio descriptions); and (2) as a source to create the corpus.

EarGram can synthesize target phrases as soon as the corpus comprises a reasonable number of units. As the earGram is synthesizing information from a live input source, the database is being created. In other words, earGram can simultaneously analyze a live input audio source and synthesize musical phrases by concatenating the audio units that are being created. This last approach is particularly suited to the practice of improvisation, because besides the automatic and meaningful segmentation of the live audio signal, the software organizes and plots the material on the screen, creating a "score" of the ongoing performance in real-time.

The visual feedback is updated on every found unit. The units' location on the screen is driven by the analysis of content according to a pre-defined descriptor space. The units' color offers a representation of the original temporal sequence of the units from a scale that goes from cold to warm colors.

The result from the collaboration between a live audio signal input and the computer response in an improvisation setup can be highly organic. An effective dialogue can be established between an ongoing performance and the computer response.

6.2.3 - SoundscapeMap

I designed the playing mode soundscapeMap to preferably manipulate and synthesize soundscapes. Targets can be defined in a similar mode as spaceMap—that is, by navigating in a two-dimensional plane. However, the plane does not provide a representation of the corpus (as in spaceMap); instead, the user navigates in a "blank" plane whose axes are assigned to musical features that are relevant to control the synthesis of soundscapes (see Figure 6.4). A particularity of this playing mode is the possibility to organize the vertical dimension of musical structure (overlapping layers of units) by the psychoacoustic notion of sensory dissonance. SoundscapeMap distributes the corpus on a maneuverable squared space according to perceptual qualities of the audio units. The navigable space is divided into four regions arranged in pairs of variables (see Figure 6.4). The first set of variables controls the density of events, and the second controls the "sharpness" of the events.

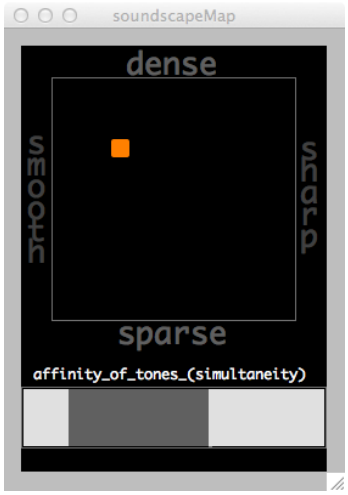


Figure 6.4 - Software interface of the playing mode soundscapeMap.

Density-sparsity regulates the number of units played simultaneously and ranges from one to five events. Smooth-sharp dichotomy, the second set of variables, controls the diversity and stability of the synthesis and is assessed by the spectral variability descriptor, which measures how quickly the non-normalized magnitude spectrum changes

over time. This descriptor not only denotes amplitude, pitch, and timbre changes in the sound, but also reveals the overall stability of the units' spectrum. Sound examples 2 and 3 were created in soundscapeMap and synthesize a target that largely goes from sparser and smoother (lower-left corner of the interface) to denser and sharper (top-right corner of the interface).

Note that the terms used in the interface are not fixed sound types; instead, they are highly dependent on the audio source(s) used to create the corpus. For instance, if we feed the system with very smooth spectral shapes as audio units, the difference between smooth and sharp will be almost imperceptible. This feature of soundscapeMap is accentuated by the need to expand the corpus through the whole plane in order to consistently explore the entire space. Otherwise, the navigation would not exploit the corpus equally, because some regions of the plane could stay empty, while other regions could contain a high density of units. The space optimization is done by equally arranging all coordinate values on both axes (see Figure 6.5 for a comparison of a corpus' distribution before and after the space optimization).

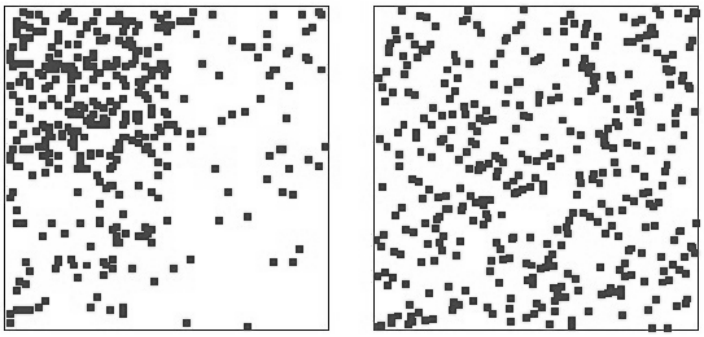


Figure 6.5 - Comparison between a corpus representation in a 2D-plot before (left image) and after (right image) space optimization.

The control of the target phrases' density in soundscapeMap offered a closer look at the quality of the resulting vertical aggregates. The density of the units itself imposes

significant changes in the quality of the "clouds of sounds" formation. A considerable amount of literature in granular synthesis is devoted to this issue (cf. Roads, 2001). However, the control of density does not provide a clear idea of the sonic characteristics of the synthesized results. The units' overlap is a quite complex tapestry with perceptual qualities shaped by a combination of factors such as the units' amplitude envelop, duration, and spectral characteristics. Therefore, the result is mostly unpredictable given the multiple factors involved.

Despite the multidimensionality of the sonic matter, or even the unpredictable factors that result from the vertical agglomeration of several units, soundscapeMap provides an extra layer of control over the vertical dimension of musical structures by regulating the sensory dissonance between overlapping units. I utilize strategies to control the vertical dimension of the synthesis in soundscapeMap instead of spaceMap—which also synthesizes vertical layers of units—due to the limited number of possible layers and the resulting transparency of the vertical aggregates in soundscapeMap.

I chose the sensory dissonance to organize the vertical layers of the synthesis because of its direct relation with the concept of harmony—a primary element of musical structure in Western music.[35] In addition, sensory dissonance provides meaningful descriptions for both pitched and non-pitched sounds, thus allowing the regulation of the sensory dissonance of both harmonic and non-harmonic sounds during generation (sound example 4 comprises three phrases separated by silence that synthesize a target that slowly change from the most consonant to the most dissonant—all remaining parameters, such as density-sparsity and smoothness-sharpness remain unchanged).

In order to control the resulting sensory dissonance of the vertical aggregates, the user must define a region of sensory dissonance in which the units should preferably fall in the interface of soundscapeMap (bottom slider of Figure 6.4). Consequently, the algorithm restricts the corpus to units that have sensory dissonance values that fall within the

[35] The reader should refer to section 3.4.2 for a detailed comparison between sensory dissonance and harmony.

selected range in relation to the last played unit. If the algorithm does not retrieve any units, it searches for the closest unit to the specified range of sensory dissonance.

6.3 - Knowledge Engineering: ShuffMeter

ShuffMeter relies on music theory knowledge to guide the generation of musical sequences that reflect a user-assigned meter. Despite the use of musical theory knowledge that is associated with the notion of style, to describe shuffMeter as a style imitation algorithm may be misleading because the meter is a musical feature that is present in centuries of musical production and associated with many musical styles. Yet, the basic principle of the algorithm can be related to algorithmic strategies for style imitation.

The generation of patterns characteristic of a meter result from the stochastic recombination of units with different stresses. The recombination in shuffMeter attempts to match a metrical template generated by Clarence Barlow's metrical indispensability algorithm (Barlow, 1987). Barlow's metric indispensability has been successfully applied as a metrical supervision procedure in the generation of drum patterns in a particular style (Bernardes et al., 2010), as well as a model for constraining a stochastic algorithm that generates rhythmic patterns given a particular time signature and metrical level (Sioros & Guedes, 2011). Before I provide a detailed description of the algorithm behind shuffMeter, one should understand Barlow's metric indispensability.

Barlow's metric indispensability algorithm defines the probabilistic weight each accent on a given meter should have in order for that meter to be perceived, that is, how indispensible each accent is at a certain metrical level for it to be felt. The accents' weights are calculated by a formula that takes into account the time signature (e.g. 4/4) and the metrical level (e.g. 16^{th} note) for which one wants to calculate the indispensabilities (see Figure 6.6). The metrical level is defined by a unique product of

prime factors, which equals the number of pulses at that metrical level and takes into account the division (binary or ternary) at higher levels. For example, the six pulses comprising the 8th note level in a meter would be defined as 3x2 (representing the three quarter notes at the quarter-note level that subdivide into two 8th notes at the level below), whereas the six pulses comprising the 8th note level in 6/8 would be represented as 2x3 (two dotted quarters that subdivide into three 8th notes). Figure 6.6 shows the normalized distribution for the 16 pulses comprising the 16th note level in 4/4.

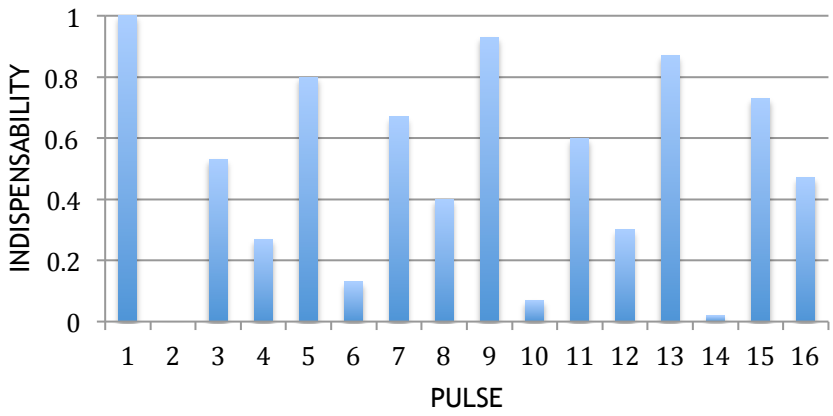


Figure 6.6 - Probability distribution given by Clarence Barlow's indispensability formula for the 16 pulses comprising the 16th note level of 4/4, which is defined as the product of prime factors 2x2x2x2.

I use Barlow's algorithm in shuffMeter to create a template that guides the definition of target phrases characteristic of a given meter. The template creation is fully automatic and relies on two parameters that must be previously assigned by the user: a time signature and a metrical level.

I ascribed the template representation to two audio descriptors: loudness and spectral variability. To simplify the computation, I merged the two descriptors into a single value by calculating their mean. In the last operation, it is assumed that spectral and loudness

changes are most likely to occur on stronger metrical accents. A similar approach is found in Sioros and Guedes (2011)—namely the use of Barlow's metric indispensability to devise a metric template whose pulses are assigned to weights according to importance in the meter so that a pattern characteristic to the meter emerges. The use of the metrical template devised by Barlow's algorithm can be also seen as a representation of the notion of phenomenal accents by Lerdahl and Jackendoff (London, 2012), that is, emphases in particular moments of the musical flow, such as dynamic accents, sudden changes in timbre, long notes, and large intervals, which contribute decisively for the perception of meter.

At each query, the algorithm retrieves the units whose mean value between the loudness and spectral variability descriptors fall on an interval defined by the indispensability value for that specific accent plus an additional range of 0.2, which results from subtracting and adding 0.1 to the indispensability value.

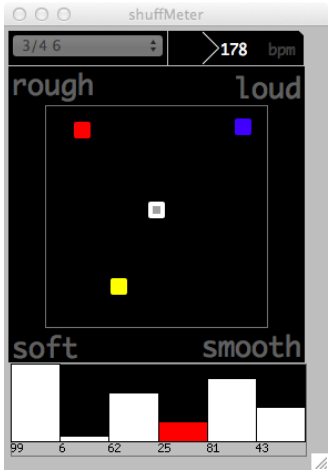


Figure 6.7 – Software interface of the playing mode shuffMeter.

Each concatenated unit is triggered by a timer that is by default assigned to a detected tempo, or manually defined by the user. Selected units whose durations do not match the specified tempo are stretched in time by using a time-stretching algorithm, which changes the speed of the audio signal without affecting the pitch. The use of a strictly timed pulse

instead of a more natural concatenation strategy results from the need to synchronize several layers of units.

The user can navigate in real time on a square present in the interface to adapt the target specifications. The navigation automatically regulates the indispensability's values. Two sets of variables mapped to each of the vertices of the square force changes to the template. Rough-smooth, will adjust the variability between all accents and loud-soft will scale the values of the template proportionally (Figure 6.8 depicts the indispensability values' distribution of each pointer present in the interface in Figure 6.7, please note the color correspondence between both figures).

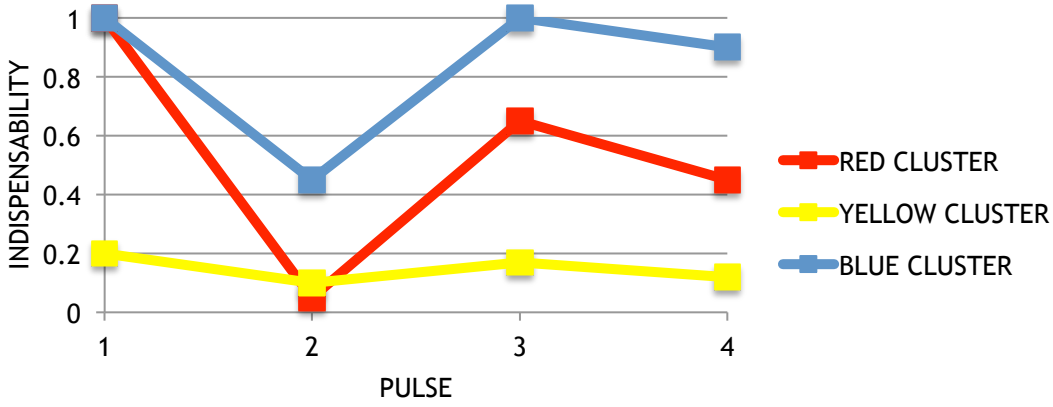


Figure 6.8 - Indispensability weights' distribution for four pulses of a 4/4 bar given by Clarence Barlow's (1987) formula. The three graphs correspond to the clusters depicted in Figure 6.7 and each configuration was scaled and conveys a percentage of variance according to their position on the navigable map.

ShuffMeter allows the creation of several synchronized layers of units. The user may consider the totality of the corpus in a single-layer recombination strategy or divide the corpus into various sub-spaces and assign each space to a different layer (sound example 5 explores the possibility to recombine and layer a collection of short drum and bass samples organized into four instrumental clusters). ShuffMeter also effectively changes the meter of a given music, and provides interesting results when clustering a corpus of

audio samples from different instruments and layering each cluster separately (sound examples 6 and 7 recombine and layer different clusters extracted from Bob Marley's *Don't Worry, Be Happy*, singed by Bobby McFerrin, utilizing two different time signatures: 3/4 and 4/4, respectively). Although the algorithm may adopt any type of unit, it conveys better results when using the beat segmentation strategy.

6.4 - Empirical Induction and Knowledge Engineering: InfiniteMode

InfiniteMode generates arbitrarily long audio streams by stretching a piece of music without affecting its tempo. The output of this mode never repeats nor loops the original audio source(s) or the new synthesized music, yet keeps playing as if on hold. I adopt two strategies in infiniteMode to extend a given musical piece. The first, structSeq, is an empirical induction method, which emulates the structure of the audio source(s) by reconstructing its time-varying morphologies. The second, chordSeq, is a knowledge engineering solution, which relies on psychoacoustic principles to guide the recombination of the corpus. InfiniteMode covers the generation of both soundscapes and polyphonic music.

From a technical standpoint, the two modes that comprise infiniteMode—structSeq and chordSeq—rely on a Markov chain algorithm for generating sequences of units. Markov chains are a special case of Markov models and a well-established algorithm to model musical data (Buys, 2011). It is one of the most popular algorithms for stochastic music generation, especially because they are very fast and easy to implement.[36]

[36] For an extensive survey of Markov processes as a compositional model please refer to Ames (1989) and Nierhaus (2009).

6.4.1 - StructSeq

StructSeq generates arbitrarily long musical excerpts, that never repeat, yet keep playing as if on hold by preserving structural elements previously extracted from a given piece of music. The algorithm attempts to emulate up to three of the following four time-varying morphologies of the audio source(s): (1) the metrical structure, and the temporal evolution of the (2) harmonic, (3) timbre, and (3) noisiness content of the audio. Earlier, I presented the process of creating representations for each of these characteristics (§ 4.1).

The user is responsible for selecting a set of characteristics adapted to the nature of the audio source(s) and application context (see Figure 6.9). In other words, the user must know if the set of chosen characteristics are meaningful in relation to the audio source(s). StructSeq's interface also suggests two generic characteristic sets adapted to the generation of soundscapes and (polyphonic) music with a strong sense of pulse. The soundscapes set includes the characteristics timbre and noisiness, and the set for polyphonic music with a strong sense of pulse includes meter, harmony, and timbre (sound example 8 utilizes structSeq to recombine and extend the initial 28 seconds of Jean-Baptiste Lully's *Les Folies d'Espagne* according to the set of characteristics suggested for polyphonic music). The target is automatically defined by the system on a unit basis, and its specifications rely on models created in advance for each characteristic involved in the generation. The selection of the best matching units is done by satisfying constraints defined hierarchically.

The temporal evolution of the timbre, noisiness, and harmony (fundamental bass) content of the audio source(s) was encoded earlier as models that specify the probabilities of transitioning between a finite set of classes for each of the characteristics. These transition probability tables serve as a basis of Markov chain algorithms, which define targets stochastically based on past events.

In order to maintain the metrical structure of the audio source(s), the algorithm

preserves sequences of units labeled consecutively according to their position in the metrical grid. For example, if a recurrent pattern of four units is found, all units in the corpus are labeled in their original order according to their position in the metrical grid, which in this case would be from zero to three. At runtime, the algorithm shuffles the units, yet preserves the sequence of numerical labels.

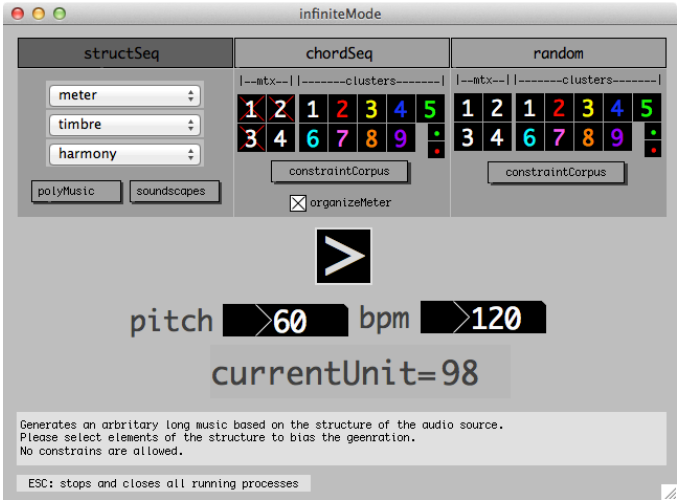


Figure 6.9 – Software interface of the playing mode infiniteMode.

The structSeq chain of operations can be described in three steps: (1) define a target specification, (2) pick the unit or collection of units that satisfies the target specification, and finally, (3) select the unit from the collection of units selected in point 2 with the most similar spectrum to the previously played unit in order to avoid discontinuities between concatenated units.

The definition of the target depends on the characteristics selected by the user in the interface and the previously played units. To define the target specification for a new unit, the algorithm examines the last played units and finds a good continuation for the selected characteristics in the interface according to the models devised during analysis. Then it retrieves all units that satisfy the requirements of each characteristic defined in the target, and finds the units that are common to all specifications. From the remaining units, it selects the one that minimizes the distance on the bark spectrum representation

to the previously selected unit.

If the algorithm does not find any unit that satisfies all the assigned characteristics, it will sequentially ignore characteristics until it finds suitable candidates. The selected characteristics on the upper slots have priority over the lower ones. If three characteristics are selected and the algorithm does not find any common units for a specific query, it will eliminate the third characteristic and again examine the number of units that satisfy the query. If it still cannot retrieve any units, it will eliminate the second characteristic.

The algorithm behind the playing mode structSeq exposes an idea suggested earlier while discussing the similarity between sound objects: the prioritization of audio features as a composition strategy. In structSeq, I use a hierarchy of audio features that is respected during unit selection, instead of assigning weights to the set of audio features. In the interface upper descriptors are satisfied at the expense of lower ones (see Figure 6.9). Therefore, the user may prioritize any modeled descriptor over others in order to achieve better (or simply different) results. Even if it is not guaranteed that the use of prioritization over weights provides more consistent results, it provides an extended and precise control to creatively explore the corpus. StuctSeq is unique because it addressed each characteristic of musical structure separately, rather than merging all parameters in a single model, like using a Hidden Markov Model for example. This last algorithm would enhance the quality of the imitation of the audio source's structure, but reduce the creative possibilities of exploring the corpus.

6.4.2 - ChordSeq

ChordSeq uses the pitch commonality model devised earlier to stochastically generate sequences of units. The strategy I applied in chordSeq was explored in the scope of generative music to create chord progressions using symbolic music representations

(Parncutt, 1991; Parncutt & Strasburger, 1994; Ferguson, 2000; Parncutt & Ferguson, 2005). Here, the concept is extended to the use of audio signals and not restricted to the generation of chord progressions.

The computation of sequences is very simple. The transition probability table of the pitch commonality between units serves as a basis of a Markov chain algorithm, which stochastically generates sequences of units. The stochastic selection of units gives preference to sequences of units with high harmonic affinity. The first unit is randomly selected among the 10 units with the highest sensory dissonance values, that is, the 10 most consonant units (sound example 9 utilizes chordSeq to recombine and extend the initial 28 seconds of Jean-Baptiste Lully's *Les Folies d'Espagne*—it is also interesting to compare sound examples 8, 9 and 10 because they recombine the same source by different generative strategies: structSeq, chordSeq, and random recombination, respectively).

The computation of pitch commonality avoids the examination of the continuity between concatenated units (i.e. concatenation cost), because the principle behind selection already includes that feature. In other words, the pitch commonality model reinforces the probability of transitioning between units whose spectrums expose similarities and continuity between units with overlapping pitches.

6.5 - Synthesis

The synthesis module is responsible for converting the information output by the playing modes into an audio signal. Synthesis also encompasses some audio effects to enhance concatenation quality and to provide greater creative expression.

The playing modes produce strings of values that convey various types of information to the synthesis module. The minimal amount of information the synthesis module may receive is a single integer, which defines the unit number to be synthesized. If no further

processing should be applied to the original raw audio data, no additional information is supplied. However, in some playing modes, such as spaceMap and shuffMeter, some additional information is compulsory. For instance, in spaceMap additional information concerning time- and frequency-shifting ratios, amplitude of the units, and spatial position should be provided. Therefore, in addition to the unit number, the output should clarify all necessary processing that must be applied to the unit. In sum, the output of the playing modes may be either a single value or a string of values, which specify additional processing.

I utilized two synthesis methods in earGram. The first method concatenates selected units with a short cross-fade. In this method, the duration of selected units is extended by 30 milliseconds (1323 samples at a 44.1kHz sample rate) to create an overlap period between adjacent units (see Figure 6.10). The second method plays selected units with a Gaussian amplitude envelope, and allows the playback of up to 200 units simultaneously.

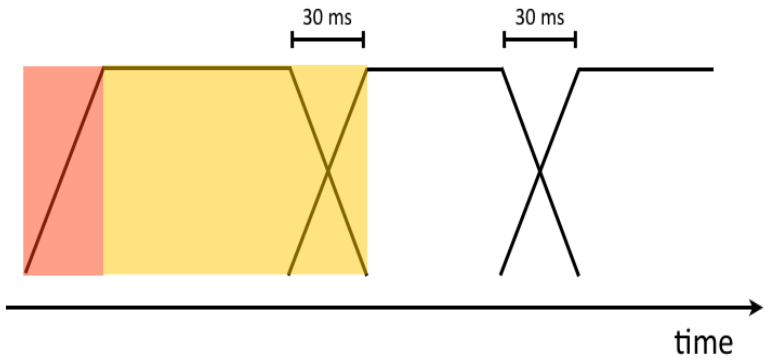


Figure 6.10 - Representation of the amplitude envelope of synthesized units with slight overlap. The yellow box corresponds to the actual duration of the unit, and the red box to the extension added to the unit in order to create the overlapping period.

The additional audio effects implemented may be divided into three categories according to their function: (1) to convey audio processing specified in the playing modes' output, (2) to allow greater artistic expression, and (3) to enhance the concatenation quality of the synthesis.

The first set of audio effects encompasses three algorithms for pitch-shifting, time-stretching, and spatializing the units. It conveys precise frequency, speed changes, and spatial position of the audio units as specified in the target phrases. This group of audio effects is frequently applied in the playing modes spaceMap and shuffMeter.

The second group allows the exploration of creative possibilities that enhance artistic expression. It comprises algorithms such as adaptive filtering, reverberation, chorus, and spectral morphing. The user may add additional effects to the available set with very little effort (in fact, all playing modes may apply this extra layer of expression).

Finally, the last category of audio effects improves the concatenation quality between adjacent units, namely by avoiding discontinuities in the spectral representation of the audio continuum. Even if most playing modes already incorporate some strategies to avoid discontinuities between adjacent units, in order to improve the concatenation quality, I added an additional feature to the end of the system to filter discontinuities in the audio spectrum. The filtering process is done by smoothing the units' transitions by creating filtering masks resulting from the interpolation of their spectra. The processing is done by an object from the Soundhack plugins bundle[37] called *+spectralcompand~*, which is a spectral version of the standard expander/compressor, commonly known as compander. It divides the spectrum in 513 bands and processes each of them individually. The algorithm computes an average of the spectrum over the last 50 milliseconds iteratively and applies it as a mask during synthesis.

[37] http://soundhack.henfast.com/.

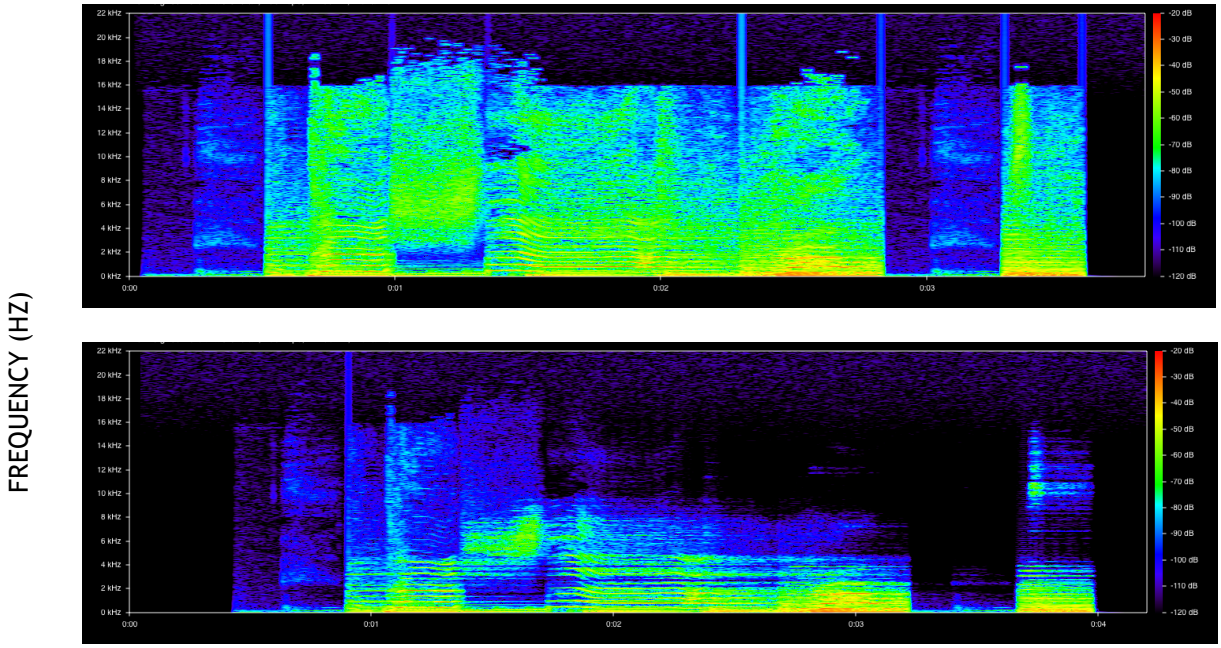


Figure 6.11 - Spectrogram representations of the same concatenated output without (top image) and with (bottom image) spectral filtering (expansion-compression).

Figure 6.11 presents two spectral analyses of four seconds of audio, which correspond to eight concatenated audio units with (top image) and without (bottom image) the processing of the spectral compander. The lower image shows a higher degree of stability and continuity between the harmonic components of the spectrum, quite noticeable in the sonic result. However, some artifacts may result from this process, such as a noticeable decrease of amplitude.

6.6 - Early Experiments and Applications of EarGram in Musical Composition

My first contact with CSS software was during the creation of the composition *In Nuce* (2011) by the Portuguese composer Ricardo Ribeiro for tenor saxophone and electronics, in

which I participated as saxophonist and musical informatics assistant. The creation of the electronic part of the composition started in 2009. Ribeiro asked me to experiment with techniques that could not only process/transform the saxophone sound, like an audio effect, such as chorus, but also to provide an extra layer of audio that could enrich and enlarge the musical gestures. I gave a single response to both of Ribeiro's requests: the adoption of CSS not only to mask the saxophone, but also to provide an extra layer of audio to enrich the timbral qualities of the piece, like a sonic transcription of the saxophone. In order to create the electronics, I started to experiment with Diemo Schwarz's CataRT and Michael Casey's SoundSpotter. After some tests, I decided to adopt SoundSpotter because of its simplicity and the possibility to work in Pure Data (the programming environment I am more familiar with). SoundSpotter offered some great results, but its "black-box" implementation offered solutions that were hard to predict and replicate.[38] From my experience, the software produces very distinct results even with the same audio signal and/or recording conditions.

Later that year, the same composer asked me to apply the same processes in a piece for ensemble and electronics, named *In Limine* (2011). From that moment on, in order to fully understand the mechanisms behind CSS and to work with more flexible solutions, I decided to start programming a small CSS patch in Pure Data. The dedicated software I built for *In Limine* enhanced the lack of predictability of SoundSpotter, and allowed me to utilize and switch between different feature spaces and experiment with different normalization strategies between input vector and corpus analysis. Later, these small patches became the core components of earGram.

Rui Dias is another Portuguese composer with whom I worked closely to utilize earGram in the creation of two of his compositions. Dias was the first composer to apply

[38] I used SoundSpotter as a Pure Data external, which allowed me to manipulate the following three parameters: (1) the number of features involved in the matching process (2) the envelope following, and (3) transition probability controls that switch between a moment-to-moment matchings and finding a location within the audio source, and bias the probability of recently played events. However, it neither allowed me to control the segmentation—the only available mode was to segment the units uniformly with durations (in samples) that needed to be necessarily a power of two—nor the quality of the audio features involved in the matching process.

earGram in a composition—*Schizophonics* (2012)—in particular to create raw material that he would later assemble in an audio sequencer. The same process was revisited a year later in an installation named *Urban Sonic Impression* (2013), whose authorship I shared with Dias. The feature that most attracted Dias in earGram was the software capability to navigate and interact with a corpus of sound units organized according to a similarity measure, which allowed him to produce granular sounds that were not possible in a practical way to do in a granulator. In both of the aforementioned pieces, Dias used the playing mode spaceMap, and the trigger mode continuousPointer, to create highly nuanced trajectories between short (200 ms) and uniform audio units.

In *Schizophonics* the use of earGram can be better understood between 4'15'' until 5'50' (sound example 11). The continuous granular layer was composer by synthesizing a trajectory drawn in the sound-space corpus visualization. The feature space used to create the corpus visualization employed weighted audio features translated to two dimensions using star coordinates. In this collaboration, and after several discussions with Dias, I realized the need to unpack the audio descriptors terminology I was using at the time—that corresponded to all descriptors from the timbreID library (Brent, 2009)—to musical jargon because the terminology adopted by most descriptors was highly inaccessible for musicians. These discussions have reinforced my motivation to redefine earGram's description scheme and resulted in the work detailed in Chapter 3.

Urban Sonic Impression is a sound installation that creates moving sound textures using sounds from the *Porto Sonoro* sound bank.[39] This work used the same playing mode as *Schizophonics*, that is, sapaceMap to create large amounts of raw material that were later edited an assembled by Dias. However, contrary to *Schizophonics*, a single audio feature—spectral brightness—was used to create the corpus representation in sound-space. Given that the navigation surface is two dimensional, both axis of the plane were assigned to the same audio feature, thus resulting in a diagonal line of ordered audio units (see Figure

[39] http://www.portosonoro.pt.

6.12). This representation allowed the creation of scales by navigating (diagonally) through the depicted line (sound examples 12 and 13). In order to fill certain "holes" in the scales and to create seamless transitions between sound units I used spectral morphing, an audio effect that has been added to earGram since then. The resulting scales were later imported into Max/MSP and used to sonify sound analysis data resultant from the project URB (Gomes & Tudela, 2013).[40] The URB data was mapped to dynamically control the reading position of the audio files generated in earGram that were being manipulated in a granulator (sound example 14 presents an excerpt of the sound installation).

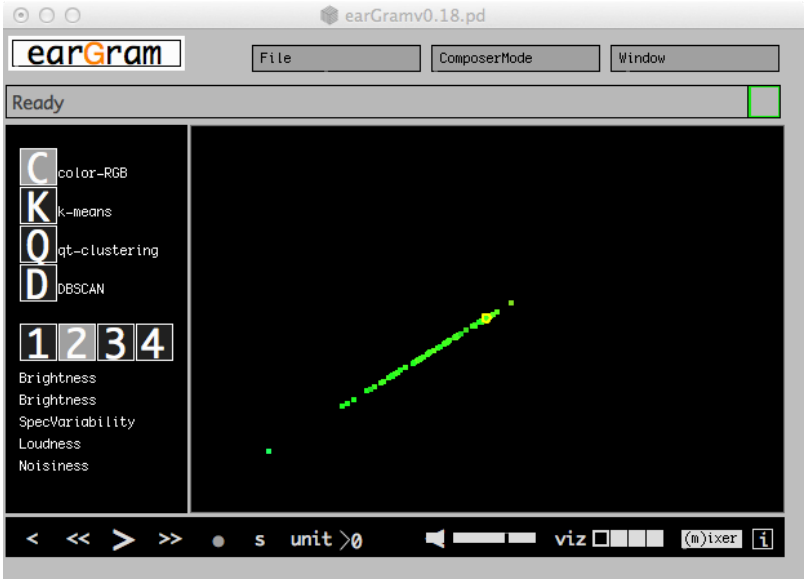


Figure 6.12 – Visualization of the corpus that supported the creation of raw material for the installation *Urban Sonic Impression*.

Nuno Peixoto was the composer who used earGram the most. Peixoto used earGram in the four following pieces: *Dialogismos I* (2012), *Dialogismos II* (2012), *Your Feet* (2012), *and A Passos de Narufágio* (2013). The first piece, *Dialogismos I*, used one of the very first versions of earGram and employed a strategy that was later even abandoned. All remaining pieces utilized more recent versions of the software and used the same strategy

[40] The URB system captures and analyzes sound features from various locations in Porto (Portugal). For more information on this project, please refer to the following web address: http://urb.pt.vu.

for composing; therefore, I decided to only detail one of them—*Your Feet*—because the material generated in earGram is exposed in the piece with extreme clarity.

The structure of *Dialogismos I* merges various elements from very different compositions. For example the pitch/harmonic structure is taken from Arvo Pärt's *Fur Alina* (1976) and the rhythmic structure from Bach's 1st Suite in G major (BWV 1007) for Unaccompanied Violoncello. This idea, and conceptual basis of the piece, relies on techniques for music appropriation/quotation by J. Peter Burckholder (1983).[41]

In *Dialogismos I*, earGram was used to synthesize the 1st Suite of Bach—encoded as MIDI files—using sound databases that include samples from Freesound,[42] and music by Wolfgang Mitterer, in particular the compositions that feature in his 2008 CD *Sopop - Believe It or Not*. The MIDI target phrases from Bach's 1st Suite were additionally filtered to only allow the synthesis of notes from particular bars of Arvo Pärt's *Für Alina*. Ultimately, the result was a tapestry of influences and a mixture of musical elements gathered from various sources. In addition to the identity of Bach and Pärt compositions, the generated music also offered a strong identification of the database sounds because they were segmented by the onset2 method in order to create sound objects that preserve the identity of the source. The strategies employed in *Dialogismos I* were not further explored and excluded from the current earGram version, mainly because I decided to focus only on the manipulation of audio signals. In addition, from all the processes Peixoto used in earGram, this was the most time-consuming and the piece that required more post-processing. The real-time capabilities of earGram are also utilized during performance in *Dialogismos I* to generate a B pedal tone (no octave is specified in the target), and target phrases encoded as MIDI information that function as interludes (or transitions) between the 6 movements of the piece.

Your Feet (sound example 15) is a paradigmatic example of the processing used in the

[41] For a deeper review of the conceptual basis of *Dialogismos I* please refer to Bernardes et al. (2012).
[42] http://www.freesound.org.

remaining pieces.[43] The most noticeable difference between the remaining abovementioned pieces of Peixoto and *Dialogismos I* is the use of an audio signal as target in the playing mode spaceMap, more specifically the liveInput trigger mode. The resulting synthesis can be seen as a sonic transcription of the target by reconstructing its morphology by other sounds (sound examples 16, 17, and 18 are "sonic transcriptions" of sound example 19, which is a MIDI synthesized version of *Your Feet* played on a piano and clarinet). After the realization of several tracks with earGram, Peixoto edited the material by layering all music generated with earGram and selecting fragments of the material.

[43] The remaining pieces can be listen in the following web address: https://sites.google.com/site/eargram/music-made-with-eargram

Chapter 7

Conclusion

In this dissertation, I formulated the hypothesis that the morphological and structural analyses of musical audio signals convey a suitable representation for computer-aided algorithmic composition, since they share the same constitutive elements manipulated through reciprocal operations. My assumptions led to the development of a framework for CAAC that manipulates representations of audio signals resulting from the structural analysis of audio sources. The ultimate aim of my work is to assist musicians to explore creative spaces, in particular to provide tools that automatically assemble sound objects into coherent musical structures according to pre-defined processes. My framework has been consequently adapted to fit the structure of a CSS algorithm and implemented as software (earGram) in the modular programing language Pure Data. EarGram, the proof-of-concept software of my analysis-synthesis framework, is a new tool for sound manipulation.

The following summary describes the steps I took in order to conceive the framework and its software implementation. Finally, I highlight the original contribution of my study

along with the artistic potential of earGram, which has been applied in several compositions that illustrate the fundamental concepts that are made possible by the software.

7.1 - Summary

The proposed framework is divided into two major modules that have a direct correspondence to the two parts that of this dissertation: analysis and composition. In Part I, I discussed and combined listening and learning competences in order to formulate a computational model to segment, describe, reveal, and model the various hierarchical layers of user-assigned audio sources.

I started by providing an overview of three musicological theories by Pierre Schaeffer, Denis Smalley, and Lasse Thoresen (Chapter 2). Special attention was given to the morphological description schemes of sound objects devised by each of the aforementioned authors. These theories in combination with MIR and psychoacoustic literature established the basis of computational strategies for segmenting an audio stream into sound objects along with their content description by a minimal and concise set of criteria (presented in Chapter 3). In turn, the description scheme provided the basis for the computation of the algorithms presented in Chapter 4 that aim to reveal and model the higher-level time scales of audio sources.

I decided to use *n*-grams to encode structural elements of sound objects because besides their reliable and simple representation of musical structural, they also provide a basis for a Markov chain algorithm, which is used here for the generation of new musical structures. There are two model groups adopted in this study. The first encodes structural elements that are extracted from the audio source(s), and the second establishes "artificial" associations between sound objects based on psychoacoustic models. The first point creates *n*-gram representations for the three following elements of musical

structure: (1) noisiness, (2) harmony, and (3) timbre. The second point encompasses two different models: (1) the probability of transitioning between all sound objects based on the affinity between sound units, and (2) the "pleasantness" between the vertical superposition of audio units. In Chapter 4, I also detailed some strategies that assist the extrapolation of higher layers of musical structure by visualizing features of the corpus organized in their original temporal order. The adopted visualization strategies are supported by audio similarity measures and clustering techniques. Finally, Chapter 4 concludes by providing two algorithms for inferring the key and meter of the audio source.

All analytical strategies devised in Part I aim not only at providing a consistent basis for the manipulation of sound objects, but also at easing the creation of sound mosaics through the automatic organization and/or recombination of their characteristics by a generative music algorithm. I examined the generation of musical structures in the second part of this dissertation.

The second part started by presenting strategies for organizing the macrostructure in earGram, followed by a description of four generative music algorithms that recombine the audio units into sequences other than their original order. The algorithmic strategies recombine the audio units by manipulating audio descriptions, and are driven by models devised during analysis—such as *n*-grams—or music theory principles. The set of algorithms presented are well known CAAC strategies, related to symbolic music representations, and cover a variety of musical contexts spanning from installations to concert music. The framework design is not constrained to any particular musical style; instead, it can be seen as an "agnostic" music system for the automatic recombination of audio tracks, guided by models learned from the audio source(s), and music and psychoacoustic theories.

7.2 - Original Contribution

This dissertation, placed at the intersection of scientific, engineering, and artistic fields, presents original contributions to each of the fields in different degrees. My artistic background provided a different perspective of, and uses for scientific findings, the application design of which was articulated and conceived through an engineering basis even though the most important application of the study resides in musical composition.

The major contribution of this study is my computational scheme for the automatic description of sound objects, regardless of the sound sources and causes. The description scheme adopts a reduced number of descriptors in comparison to analogous state-of-the-art applications. However, it covers the most prominent classes of relatively independent audio descriptors from a statistical point of view, and presents low levels of information redundancy.

Even though I relied on MIR research to mathematically define the audio descriptors, the computation has been subject to small adjustments. For example, the noisiness descriptor is computed by a weighted combination of low-level audio features that balances the characterization between pitched and noisy sounds, and encompasses more subtleties that are hardly expressed by a single low-level audio feature. Furthermore, in order to adopt a uniform scale for all descriptors—a feature that is compulsory in many MIR applications—the description scheme adopts specific scaling factors in relation to the descriptor at issue. The uniform range of the descriptors' output avoids the need to normalize the data by any statistical feature, which normally leads to a consequent lack of meaning in the results.

The last innovative aspect of the description scheme I would like to highlight is the implementation of psychoacoustic dissonance models as audio descriptors. The use of such descriptors offers a systematic characterization of the harmonic timbre qualities of the sound objects and allows the creation of probabilistic models that can ultimately guide

the generation of transitions between sound objects and/or their overlap.

Based on the low-level descriptions of the units, mid- and high-level representations of the corpus are inferred and/or presented to the user in an intuitive manner through the use of visuals in earGram. Although the algorithms used for the mid-level description of the corpus are not original contributions, their articulation in a single framework is unique. The corpus visualizations represent the higher layers of the audio source's structure, which facilitates the reorganization and/or exploration of smaller sections of the source during generation. In particular, by depicting and grouping the sound objects that compose the corpus according to their similarity it is possible to expose the main characteristics of the corpus. In addition, if the sound objects are organized in their original temporal order—as in the self-similarity matrix—it is even likely to get an idea of the macrostructure of the audio source(s). Based on this information, the user may constrain the corpus to smaller sections that expose particular characteristics and use them differently while composing with earGram.

The aim behind the analytical strategies implemented in earGram is the exploration of the corpus by generative music strategies. In fact, the analytical tools were shaped to convey easy and fast experimentation of sound objects mosaics for musicians. Even though the analytical tools implemented in earGram suggest their suitability for generative music purposes, the following paragraphs will illustrate how the sound objects' representations devised during analysis are manipulated to generate consistent music results.

The generative strategies implemented in earGram allow the manipulation of several hierarchical layers of musical structure, with different degrees of automation. For example, while the user needs to manually assign the subsets of the corpus used in each section or phrase, the low-level selection of sound objects is entirely automatic and managed by the system according to user-given specifications.

The possibilities offered by earGram to create sub-corpora of units minimize a major drawback of most generative music strategies, that is, the organization of the meso and

macro levels of musical structure. Interestingly, the adopted principle for organizing the meso and macro structure in earGram follows the same principle as the method for assembling the low-level morphology of the music surface (at the sound object time scale), that is, through the use of selection principles.

As far as the low-level units selection in earGram is concerned, the units' descriptions were successfully applied as sound objects' representations in known CAAC strategies, such as Markov chains, tendency masks, or rule-based algorithms. The adopted units' representations solved the problem of low-level information, complexity, and density of audio signals that make them extremely difficult to manipulate in generative music. The following four generative music strategies were developed and implemented in earGram as unit selection algorithms of a CSS system: (1) spaceMap, (2) soundscapeMap, (3) shuffMeter, and (4) infiniteMode. The four playing modes encompass very distinct strategies for generating music, spanning from micromontage for sound design to more traditional generative approaches for polyphonic music, such as algorithms for style imitation and/or the emulation of music theory principles.

In terms of creative output, the four generative strategies implemented in earGram not only provide the composer tools for the fast and easy creation of large amounts of raw material for a particular composition, but also allow more ready-to-use solutions that can actively participate in live performances. In the first point, earGram can be seen as a computer-aided composition system, in a similar fashion as improvisation may serve the composer for the preliminary exploration of an idea and/or to create large chunks of raw material that can be manually assembled later. Concerning the second point, the playing modes implemented in earGram were designed to consistently produce results according to pre-defined processes or to explore, manipulate, and interact with a corpus of sound objects in real-time—particularly by navigating in spaces that define and/or constrain the generation of target phrases to be synthesized.

The first compositional feature implemented in earGram that I would like to highlight

is the possibility to systematically work with audio features like noisiness, width, and brightness that are commonly understood as secondary elements of musical structure in Western music. In other words, one may work outside of the pitch-duration primacy because the description scheme devised in combination with the generative strategies allow the systematic and identical manipulation of all descriptors. For example, spaceMap synthesizes target phrases that are drawn on top of a corpus visualization organized by audio features. Thus, the user can synthesize trajectories by navigating in a visualization organized according to sensory dissonance and noisiness, or any other combination of descriptors.

Still, with regard to spaceMap, it is interesting to note that the definition of target phrases by drawing trajectories on the interface offers interesting avenues for composition—especially if one explores the trajectories as musical gestures/events. One can build compositional systems based on visual trajectories. If the same trajectory is drawn with distinct audio feature spaces, it is possible to create a sort of sonic transcription and/or variation of the same "musical" gesture. Also, one may create visually related gestures (i.e. mirrored, inverse, etc.) that somehow expose sonic affinities.

SoundscapeMap follows the same mode of interaction as spaceMap, that is, the definition of targets through physical navigation in a constricted space on the interface, but the possibilities to guide generation are directed toward soundscapes. SoundscapeMap exposes how CSS may be ideal to procedural audio applications. One of the most innovative aspects of soundscapeMap is its ability to organize the sensory dissonance of vertical aggregates of musical structure, a feature commonly overlooked in CSS.

The two remaining playing modes—shuffMeter and infiniteMode—focus on more traditional music making strategies. Still, it presents solutions for dealing with audio that would take considerable time and effort if done manually. For instance, shuffMeter allows rapid experimentation with different time signatures and various layers using the same

audio source(s).

ShuffMeter utilizes a strategy that may be ideal for composing with commercial audio loops clustered by instruments in order to create cyclic patterns for each instrument (defined by the metric structure), which can be varied by navigating in a simple interface—an ideal tool for practices such as DJing. Finally, infiniteMode limitlessly extends a given audio source by preserving the structural characteristics of the audio source(s), yet reshuffling the original order of its constituent sound objects. In addition, it is possible to experiment with sound object progressions based on the affinity of tones between sound objects, referred to as pitch commonality in psychoacoustics. InfiniteMode also allows the specification and prioritization of the characteristics to guide the generation. In other words, one can use this playing mode to slightly alter the morphology of the source by changing the prioritization and/or the features involved in the generation. For instance, one may only reshuffle a particular audio track by preserving its metric structure and ignoring all other components.

After addressing the creative uses of earGram, I would like to make a few remarks on the technical basis of earGram; in particular, the means by which this study extends CSS, even as a consequence of the adopted methodology, since it was not intended as a primary objective. The extensive use and implications of this synthesis technique led me to examine in detail its technical and conceptual basis. Therefore, many considerations present in this dissertation may decisively contribute to the development of this synthesis technique.

The particular innovative aspect of earGram in relation to other CSS is the use of generative music strategies as unit selection algorithms, as opposed to finding the best candidate unit to a target representation based on the similarity between *n*-dimensional feature vectors. Additionally, the database in earGram is understood as a time-varied resource in a system that allows the user to dynamically assign sub-spaces of the corpus that are easily interchangeable at runtime. Therefore, there is no pre-defined or fixed set

of audio feature vectors. Instead, the system is highly flexible and explores weighting, prioritizing, and constraining audio features adapted to particular audio sources and application contexts. Finally, despite the common synthesis of overlapping audio units in CSS, its organization is commonly overlooked. I proposed the use of psychoacoustic dissonance models, in particular sensory dissonance, to examine and consequently organize the vertical dimension of musical structure.

While the analytical part of the model described in this dissertation relied heavily on music and musicological theory, many decisions taken in its generative counterpart relied heavily on empirical judgments. The readers may judge for themselves the quality of the results produced by the system by listening to some sound examples at: https://sites.google.com/site/eargram/ (also included in the accompanying CD), and tryout the software with various playing modes and different sound corpora.

The musical examples made available not only testify the effectiveness of the system, but also illustrate the artistic potential of the detailed CAAC methods. In addition, collaborations with three Portuguese composers—Ricardo Ribeiro, Rui Dias, and Nuno Peixoto—in eight compositions have both tested and verified the usefulness of earGram, and contributed actively to the software's design.[44]

7.3 - Future Work

I have designed the four recombination modes detailed here to not only assist the composer at work by providing him/her raw material, but also to participate in live performances. Despite the real-time capabilities of the system, its effective contribution to a live performance can be enhanced if prior experimentation and organization of the material has been made. However, the user may need to limit the corpus to a collection

[44] One of the collaborations with the composer Nuno Peixoto has been reported in a peer-reviewed paper presented at the ARTECH 2012 – 6th International Conference on Digital Arts (Bernardes et al., 2012). For a complete list of compositions created with earGram please refer to Appendix C.

of units that more effectively generate coherent music results. This experimentation phase could be avoided if the system had more high-level information concerning the sound source(s) and greater knowledge of the structural function of each unit in the overall composition of the audio source(s).

Another particularity of the framework that could enhance the synthesis results is the adoption of categorical descriptions grounded in perceptual sound qualities. In particular, the aspect that would greatly profit from the use of perceptual categories of sound would be the modeling strategies of the current framework. In order to divide the audio features at issue into sound typologies to model the temporal evolution of particular musical elements, I divided the descriptor's range into an arbitrary number of categories. The range of each category ("sound typology") is artificial and does not take into account any musical or perceptual considerations.

The counterpart of the framework—composition—may adopt audio effects at the end of the processing chain to enhance the concatenation quality and provide greater expressivity. Integrating more audio effects into the framework could expand its possibilities. In addition to this, the use of audio effects could fill some gaps in the database. In other words, instead of simply finding the best matching units for a particular target specification, the system could apply transformations that would provide better matches.

Finally, although the main purpose behind the listening and learning modules and the visualization strategies is to drive synthesis, its range of application could be expanded toward areas such as musical analysis—namely computational musicology and cognitive musicology.

Bibliography

Altmann, P. (1977). *Sinfonia von Luciano Berio: Eine analytische studie*. Vienna: Universal Edition.

Ames, C. (1989). The Markov process as a compositional model: A survey and tutorial. *Leonardo, 22*(2), 175-187.

Apel, K. (1972). *Harvard dictionary of music*. (2nd ed.). Cambridge, MA: Harvard University Press.

Ariza, C. (2004). An object-oriented model of the Xenakis sieve for algorithmic pitch, rhythm, and parameter generation. *Proceedings of the International Computer Music Conference*, 63-70.

Ariza, C. (2005). Navigating the landscape of computer-aided algorithmic composition systems: A definition, seven descriptors, and a lexicon of systems and research. *Proceedings of the International Computer Music Conference*, 765-772.

Aucouturier, J.-J. and Pachet, F. (2005). Ringomatic: A Real-Time Interactive Drummer Using Constraint-Satisfaction and Drum Sound Descriptors. *Proceedings of the International Conference on Music Information Retrieval*, 412-419.

Barlow, C. (1980). Bus journey to parametron. *Feedback Papers*, *21-23*. Cologne: Feedback Studio Verlag.

Barlow, C. (1987). Two essays on theory. *Computer Music Journal*, *11*(1), 44-60.

Bello, J. P., & Sandler, M. (2003). Phase-based note onset detection for music signals. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, *5*, 441-444.

Bello, J. P., Duxbury, C., Davies, M. E., & Sandler, M. B. (2004). On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, *11*(6), 553-556.

Bennett, R. (1981). *Form and design*. Cambridge: Cambridge University Press.

Bernardes, G., Guedes, C., & Pennycook, B. (2010). Style emulation of drum patterns by means of evolutionary methods and statistical analysis. *Proceedings of the Sound and Music Computing Conference*.

Bernardes, G., Peixoto de Pinho, N., Lourenço, S., Guedes, C., Pennycook, B., & Oña, E. (2012). The creative process behind *Dialogismos I*: Theoretical and technical considerations. *Proceedings of the ARTECH 2012—6th International Conference on Digital Arts*, 263-268.

Bernardes, G., Guedes, C., & Pennycook, B. (2013). EarGram: An application for interactive exploration of concatenative sound synthesis in Pure Data. In M. Aramaki, M. Barthet, R. Kronland-Martinet, & S. Ystad (Eds.), *From sounds to music and emotions* (pp. 110-129). Berlin-Heidelberg: Springer-Verlag.

Beyls, P. (1989). The musical universe of cellular automata. *Proceedings of the International Computer Music Conference*, 34-41.

Bidlack, R. (1992). Chaotic systems as simple (but complex) compositional algorithms. *Computer Music Journal*, *16*(3), 33-47.

Brent, W. (2009). A timbre analysis and classification toolkit for Pure Data. *Proceedings of the International Computer Music Conference*.

Brent, W. (2011). A perceptually based onset detector for real-time and offline audio parsing. *Proceedings of the International Computer Music Conference*.

Brossier, P. (2006). *Automatic annotation of musical audio for interactive applications.* PhD dissertation, Centre for Digital Music, Queen Mary University of London.

Burkholder, J. P. (1983). *The evolution of Charles Ives's music: Aesthetics, quotation, technique*. PhD dissertation, University of Chicago.

Burkholder, J. P. (1994). The uses of existing music: Musical borrowing as a field. *Notes*, *50*(3), 851-870.

Buys, J. (2011). *Generative models of music for style imitation and composer recognition*. Honouros project in computer science, final report, University of Stellenbosch. Retrieved August 12, 2013, from http://www.cs.sun.ac.za/rw778/files/2011/11/j_buys_hons_report_2011.pdf.

Cage, J. (1962). *Werkverzeichnis*. New York, NY: Edition Peters.

Caires, C. (2004). IRIN: Micromontage in a graphical sound editing and mixing tool. *Proceedings of the International Computer Music Conference*.

Cambouropoulos, E. (1994). Markov chains as an aid to computer-assisted composition. *Musical Praxis, 1*(1), 41-52.

Cano, P., Koppenberger, M., Wack, N., Garcia, J., Masip, J., Celma, O., Garcia, D., Gómez, E., Gouyon, F., Guaus, E., Herrera, P., Massaguer, J., Ong, B., Ramirez, M., Streich, S., & Serra, X. (2005). An industrial-strength content-based music recommendation system. *Proceedings of the International ACM SIGIR Conference*.

Cardle, M., Brooks, S., & Robinson, P. (2003). Audio and user directed sound synthesis. *Proceedings of the International Computer Music Conference*.

Casey, M. (2009). Soundspotting: A new kind of process? In R. Dean (Ed.), *The Oxford handbook of computer music*. New York, NY: Oxford University Press.

Chafe, C., Mont-Reynaud, B., & Rush, L. (1982). Toward an intelligent editor of digital audio: Recognition of musical constructs. *Computer Music Journal*, *6*(1), 30–41.

Chai, W. (2005). *Automated analysis of musical structure*. PhD dissertation, Massachusetts Institute of Technology.

Chion, M. (1983). *Guide des objets sonores: Pierre Schaeffer et la recherche musicale*. Paris: INA/Buchet-Chastel.

Christensen, E. (2004). Overt and hidden processes in 20th century music. *Axiomathes*, *14*(1), 97-117.

Conklin, D., & Witten, I. H. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research*, *24*(1).

Connell, J. (2011). *Musical mosaicing with high level descriptors*. Master thesis, Pompeu Fabra University.

Cont, A. (2008). *Modeling musical anticipation: From the time of music to the music of time*. PhD dissertation, University of California at San Diego.

Cooper, G., & Meyer, L. (1960). *The rhythmic structure of music*. Chicago, IL: University of Chicago Press.

Cope, D. (1984). *New directions in music*. Dubuque, IA: W. C. Brown.

Cope, D. (1993). Virtual music. *Electronic Musician*, *9*(5), 80-85.

Cope, D. (1996). *Experiments in musical intelligence*. Madison, WI: A-R Editions.

Cope, D. (2001). *Virtual music: Computer synthesis of musical style*. Cambridge, MA: The MIT Press.

Cox, C., & Warner, D. (Eds.). (2004). *Audio culture: Readings in modern music*. New York, NY: Continuum International Publishing Group.

Csikszentmihalyi, M. (2009). *Creativity: Flow and the psychology of discovery and invention*. New York, NY: HarperCollins.

Cutler, C. (2004). Plunderphonia. In C. Cox & D. Warner (Eds.), *Audio culture: Readings in modern music* (pp. 138-156). New York, NY: Continuum International Publishing Group.

Davies, M., & Plumbley, M. (2006). A spectral difference approach to downbeat extraction in musical audio. *Proceedings of the 14th European Signal Processing Conference*.

Davies, M., & Plumbley, M. (2007). Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(3), 1009-1020.

Delalande, F. (1998). Music analysis and reception behaviours: Sommeil by Pierre Henry. *Journal of New Music Research, 27*(1-2).

Di Scipio, A. (2005). Formalization and intuition in Analogique A et B. *Definitive Proceedings of the International Symposium Iannis Xenakis*.

Di Scipio, A. (2009). Micro-time sonic design and timbre formation. *Contemporary Music Review*, *10*(2), 135-148.

Dixon, S. (2006). Onset detection revisited. *Proceedings of the International Conference on Digital Audio Effects*, 133–137.

Downie, J. S. (2004). The scientific evaluation of music information retrieval systems: Foundations and future. *Computer Music Journal*, *28*(2), 12-23.

DuBois, L. (2003). *Applications of generative string-substitution systems in computer music*. PhD dissertation, Columbia University.

Dunn, P. (2005). *Measurement and data analysis for engineering and science*. New York, NY: McGraw–Hill.

Eigenfeldt, A. (2009). The evolution of evolutionary software intelligent rhythm generation in kinetic engine. *Proceedings of EvoMusArt'09, the European Conference on Evolutionary Computing*.

Ellis, D., Whitman, B., Berenzweig, A., & Lawrence, S. (2002). The quest for ground truth in musical artist similarity. *Proceedings of the International Conference on Music Information Retrieval*.

Ellis, D. (2007). Beat tracking by dynamic programming. *Journal of New Music Research*, *36*(1), 51-60.

Eronen, A. (2001). Comparison of features for musical instrument recognition. *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*.

Essl, K. (1995). Lexikon-Sonate. An interactive realtime composition for computer-controlled piano. *Proceedings of the II Brazilian Symposium on Computer Music*.

Ester, M., Kriegel H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Knowledge Discovery and Data Mining*, *96*, 226–231.

Farbood, M., & Schoner, B. (2001). Analysis and synthesis of Palestrina-style counterpoint using Markov chains. *Proceedings of the International Computer Music Conference*, 471-474.

Ferguson, S. (2000). *Concerto for piano and orchestra*. PhD dissertation, McGill University.

Fineberg, J. (2000). Spectral music. *Contemporary Music Review*, *19*(2), 1-5.

Foote, J. (1999). Visualizing music and audio using self-similarity. *Proceedings of ACM Multimedia*.

Frisson, C., Picard, C., Tardieu, D., & Pl-area, F. R. (2010). Audiogarden: Towards a usable tool for composite audio creation. *Quarterly Progress Scientific Reports of the Numediart Research Program*, *3*(2), 33-36.

Gainey, C. (2009). *Turning sound into music: Attitudes of spectralism*. PhD dissertation, The University of Iowa.

Geiringer, K. (1950). Artistic interrelations of the Bachs. *The Musical Quarterly*, *36*, 363-374.

Gomes, J., & Tudela, D. (2013). Urb: Urban sound analysis and storage project. *Proceedings of the Sound and Music Computing Conference*.

Gómez, E. (2005). Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing, 17*(11).

Gómez, E., & Bonada, J. (2005). Tonality visualization of polyphonic audio. *Proceedings of the International Computer Music Conference*.

Gómez, E. (2006a). Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing, Special Cluster on Computation in Music*, *18*(3).

Gómez, E. (2006b). *Tonal Description of Music Audio Signals*. PhD dissertation, Pompeu Fabra University.

Goto, M., & Muraoka, Y. (1998). Music understanding at the beat level: Real-time beat tracking for audio signals. *Proceedings of IJCAI-95 Workshop on Computational Auditory Scene Analysis*, 157-176.

Goto, M. (2001). An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, *30*(2), 159-171.

Goto, M. (2003). A chorus-section detecting method for musical audio signals. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 14*(5).

Gouyon, F., & Herrera, P. (2003). Determination of the meter of musical audio signals: Seeking recurrences in beat segment descriptors. *114th Audio Engineering Society Convention*.

Gouyon, F., & Dixon, S. (2005). A review of automatic rhythm description systems. *Computer Music Journal, 29*(1), 34-54.

Gouyon, F., Herrera, P., Gómez, E., Cano, P., Bonada, J., Loscos, A., Amatriain, X., & Serra, X. (2008). Content processing of music audio signals. In P. Polotti & D. Rocchesso (Eds.), *Sound to sense, sense to sound: A state of the art in sound and music computing* (pp. 83-160). Berlin: Logos Verlag Berlin GmbH.

Grachten, M., Schedl, M., Pohle, T., & Widmer, G. (2009). The ISMIR cloud: A decade of ISMIR conferences at your fingertips. *Proceedings of the International Conference on Music Information Retrieval*.

Griffiths, P. (1981). *Modern music: The avant-garde since 1945*. New York, NY: George Braziller.

Hackbarth, B., Schnell, N., & Schwarz, D. (2010). *Audioguide: A framework for creative exploration of concatenative sound synthesis*. Musical Research Residency Report, IRCAM, Paris.

Hackbarth, B., Schnell, N., Esling, P., & Schwarz, D. (2013). Composing morphology: Concatenative synthesis as an intuitive medium for prescribing sound in time. *Contemporary Music Review, 32*(1), 49-59.

Hazel, S. (2001). *Soundmosaic*. Web page. Retrieved July 24, 2013, from http://thalassocracy.org/soundmosaic.

Herrera, P., Serra, X., & Peeters, G. (1999). Audio descriptors and descriptor schemes in the context of MPEG-7. *Proceedings of the International Computer Music Conference*.

Herrera, P., Dehamel, A. & Gouyon, F. (2003). Automatic labeling of unpitched percussion sounds. *Proceedings of the 114th Audio Engineering Society Convention*.

Heyer, L., Kruglyak S., & Yooseph, S. (1999). Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research, 9*(11), 1106-1115.

Hild, H., Feulner, J., & Menzel, W. (1992). HARMONET: A neural net for harmonizing chorales in the style of J.S. Bach. In R. Lippmann, J. Moody, & D. Touretzky (Eds.), *Advances in neural information processing 4* (pp. 267-274). Morgan Kaufmann.

Holm-Hudson, K. (1997). Quotation and context: Sampling and John Oswald's plunderphonics. *Leonardo Music Journal*, *7*, 17-25.

Hoskinson, R., & Pai, D. (2001). Manipulation and resynthesis with natural grains. *Proceedings of the International Computer Music Conference*.

Hunt, A., & Black, A. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 373-376.

Inselberg, A. (2009). *Parallel coordinates: Visual multidimensional geometry and its applications*. Berlin: Springer-Verlag.

Izmirli, O. (2005). Template based key finding from audio. *Proceedings of the International Computer Music Conference*.

Jacob, B. L. (1996). *Algorithmic composition as a model of creativity*. Retrieved April 24, 2013, from http://www.ee.umd.edu/~blj/algorithmic_composition/algorithmicmodel.html.

Jehan, T. (2005). *Creating music by listening*. PhD dissertation, Massachusetts Institute of Technology.

Jehan, T. (2010). *US Patent No. 7842874 B2*. Washington, DC: U.S. Patent and Trademark Office.

Jensen, K. (1999). *Timbre models of musical sounds*. Doctoral dissertation, Department of Computer Science, University of Copenhagen.

Jurafsky, D., Martin, J., Kehler, A., Vander Linden, K., & Ward, N. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (Vol. 2). Upper Saddle River: Prentice Hall.

Kandogan, E. (2000). Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. *Proceedings of the IEEE Information Visualization Symposium*.

Kandogan, E. (2001). Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 107-116.

Kim, H.-G., Moreau, N., & Sikora, T. (2005) *MPEG-7 Audio and beyond: Audio content indexing and retrieval*. Chichester, UK: John Wiley & Sons.

Klapuri, A. (2003). Musical meter estimation and music transcription. *Proceedings of the Cambridge Music Processing Colloquium*.

Klapuri, A., Eronen, A., & Astola, J. (2006). Analysis of the meter of acoustic musical signals. *IEEE Transactions on Speech and Audio Processing*.

Knuth, D. (1968). *The art of computer programming: Fundamental algorithms* (vol. 1). Addison-Wesley.

Kobayashi, R. (2003). Sound clustering synthesis using spectral data. *Proceedings of the International Computer Music Conference*.

Koenig, G. M. (1978). Composition processes. In M. Battier & B. Truax (Eds.), *Computer music*. Canadian Commission for UNESCO.

Krimphoff J., McAdams, S., & Winsberg, S. (1994). Caractérisation du timbre des sons complexes. II Analyses acoustiques et quantification psychophysique. *Journal de Physique IV*, 4(C5), C5-625.

Krumhansl, C. (1990). *Cognitive foundations of musical pitch*. New York, NY: Oxford University Press.

Krumhansl, C. (2005). The geometry of musical structure: A brief introduction and history. *Computers in Entertainment (CIE)*, *3*(4), 1-14.

Landy, L. (2007). *Understanding the art of sound organization*. Cambridge, MA: The MIT Press.

Lannes, Y. (2005). *Synthèse de la parole par concaténation d'unités*. Master thesis, Université Toulouse III Paul Sabatier.

Lazier, A., & Cook, P. (2003). MOSIEVIUS: Feature driven interactive audio mosaicing. P*roceedings of the International Conference on Digital Audio Effects*.

Leach, J., & Fitch, J. (1995). Nature, music, and algorithmic composition. *Computer Music Journal*, *19*(2).

Lee, C. S. (1991). The perception of metrical structure: Experimental evidence and a model. In P. Howell, R. West, & I. Cross (Eds.), *Representing musical structure*. London: Academic Press.

Lerdahl, F., & Jackendoff, R. (1983). *A generative theory of tonal music*. Cambridge, MA: The MIT Press.

Lessig, L. (2001). *The future of ideas: The fate of the commons in a connected world*. New York, NY: Random House.

Lessig, L. (2004). *Free culture: How big media uses technology and the law to lock down culture and control creativity*. New York, NY: The Penguin Press.

Lessig, L. (2008). *Remix: Making art and commerce thrive in the hybrid economy*. New York, NY: The Penguin Press.

Leung, T.-W. (2008). *Memory, aesthetics and musical quotation: Four case studies in 20th century music*. Master thesis, The University of Hong Kong.

Lindemann, E. (2001). *Musical synthesizer capable of expressive phrasing*. US Patent 6, 316, 710.

London, J. (2012). *Hearing in time: Psychological aspects of musical meter*. New York, NY: Oxford University Press.

Longuet-Higgins, H., & Lee, C. (1982). Perception of musical rhythms. *Perception*, *11*(2), 115-128.

Loy, D. (1985). Musicians make a standard: The MIDI phenomenon. *Computer Music Journal*, *9*(4).

Lu, L., Jiang, H., & Zhang, H.-J. (2001). Robust audio classification and segmentation method. *Proceedings of the ACM International Multimedia Conference and Exhibition*, 103–211.

MacCallum, J., & Einbond, A. (2008). Real-time analysis of sensory dissonance. In R. Kronland-Martinet, S. Ystad, & K. Jensen (Eds.), *Computer music modeling and retrieval: Sense of sounds* (pp. 203-211). Berlin-Heidelberg: Springer-Verlag.

MacKay, D. (2003). *Information theory, inference and learning algorithms*. Cambridge: Cambridge University Press.

Macon, M. W., Cronk, A. E., & Wouters, J. (1998). Generalization and discrimination in tree-structured unit selection. *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*.

Malloch, J. (2005). *Beat and tempo induction for musical performance*. Technical Report MUMT-IDMIL-05-01, Music Technology Area, McGill University.

Manion, M. (1992). *From tape loops to MIDI: Karlheinz Stockhausen's forty years of electronic music*. Retrieved July 23, 2012, from http://www.stockhausen.org/tape_loops.html.

Martin, A. (2011). *Touchless gestural control of concatenative sound synthesis*. Master thesis. Schulich School of Music, McGill University, Montreal, Canada.

Martin, K. (1999). *Sound source recognition: A theory and computational model*. PhD dissertation, Massachusetts Institute of Technology.

Meron, Y. (1999). *High quality singing synthesis using the selection-based synthesis scheme*. PhD dissertation, University of Tokyo.

Mikula, L. (2008). *Concatenative music composition based on recontextualisation utilising rhythm-synchronous feature extraction*. Diploma Thesis, Institute of Electronic Music and Acoustics, University of Music and Dramatic Arts, Graz, Austria.

Miller, P. (2004). *Rhythm science*. Cambridge, MA: The MIT Press.

Miller, P. (Ed.). (2008). *Sound unbound: Sampling digital music and culture*. Cambridge, MA: The MIT Press.

Miranda, E. R. (2001). Evolving cellular automata music: From sound synthesis to composition. *ALMMA 2001: Proceedings of the Workshop on Artificial Life Models for Musical Applications*.

Misdariis, N., Smith, B., Pressnitzer, D., Susini, P., & McAdams, S. (1998). Validation of a multidimensional distance model for perceptual dissimilarities among musical timbres. *Proceedings of 135th Meeting of the Acoustical Society of America / 16th International Congress on Acoustics*.

Mitrovic, D., Zeppelzauer, M., & Eidenberger, H. (2006). Analysis of the data quality of audio descriptions of environmental sounds. *Fourth Special Workshop Proceedings*, 70-79.

Moore, F. R. (1988). The dysfunctions of MIDI. *Computer Music Journal*, *12*(1), 19-28.

Mozer, M. C. (1994). Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing. *Connection Science*, *6*(2-3), 247-280.

Nierhaus, G. (2009). *Algorithmic composition: Paradigms of automatic music generation*. Vienna: Springer-Verlag.

Norowi, N. M., & Miranda, E. R., (2011). Order dependent feature selection in concatenative sound synthesis using analytical hierarchy process. *Proceedings of the EUROCON—International Conference on Computer as a Tool,* 1-4.

Oliveira, J. L., Gouyon, F., Martins, L. G., & Reis, L. P. (2010). IBT: A real-time tempo and beat tracking system. *Proceedings of the 11th International Conference on Music Information Retrieval*.

Ong, B. S., & Herrera, P. (2005). Semantic segmentation of music audio. *Proceedings of the International Computer Music Conference*.

Oswald, J. (1986). Plunderphonics, or audio piracy as a compositional prerogative. *Musicworks*, 5-8.

Oswald, J. (2001). *Plunderphonics 69/96.* [CD]. Fony/Seeland.

Ovans, R., & Davison, R. (1992). An interactive constraint-based expert assistant for music composition. *9th Canadian Conference on Artificial Intelligence,* 76-81.

Pachet, F. and Roy, P. (2001). Musical harmonization with constraints: A survey. Constraints, *6*(1), 7-19.

Palombini, C. (1993). Machine songs V: Pierre Schaeffer: From research into noises to experimental music. *Computer Music Journal, 17*(3), 14-19.

Parncutt, R. (1989). *Harmony: A psychoacoustical approach*. Berlin: Springer-Verlag.

Parncutt, R. (1991). A psychoacoustical model of tonal composition. *Proceedings of the International Computer Music Conference*.

Parncutt, R., & Strasburger, H. (1994). Applying psychoacoustics in composition: "Harmonic" progressions of "non-harmonic" sonorities. *Perspectives of New Music*, *32*(2), 1-42.

Parncutt, R., & Ferguson, S. (2005). Comporre con l'apporto di un algoritmo di teoria della percezione (computer-assisted composition with algorithmic implementations of pitch-perceptual theory). *Rivista di Analisi e Teoria Musicale*, *10*(2), 103-118.

Paulus, J., Müller, M., & Klapuri, A. (2010). State of the art report: Audio-based music structure analysis. *Proceedings of the International Conference on Music Information Retrieval*, 625-636.

Pauws, S. (2004). Musical key extraction from audio. *Proceedings of the International Conference on Music Information Retrieval*.

Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the cuidado project. Ircam, Cuidado Project Report.

Peeters, G. (2006). Chroma-based estimation of musical key from audio-signal analysis. *Proceedings of the International Conference on Music Information Retrieval*, 115-120.

Peeters, G., & Deruty, E. (2008). Automatic morphological description of sounds. *Acoustics 2008,* Paris, France.

Peeters, G., Giordano, B.L., Susini, P., Misdariis, N., & McAdams, S. (2011). The timbre toolbox: Extracting audio descriptors from musical signals. *Journal of Acoustical Society of America*, *130*(5), 2902-2916.

Porres, A. (2011). Dissonance model toolbox in Pure Data. *Proceedings of the 4th Pure Data Convention*.

Porres, A. (2012). *Modelos psicoacústicos de dissonância para electrônica ao vivo*. PhD dissertation. Escola de Comunicações e Artes, Universidade de São Paulo, Brazil.

Povel, D., & Essens, P. (1985). Perception of temporal patterns. *Music Perception*, *2*(4), 411-440.

Pressing, J. (1988). Score generation with L-systems. *Proceedings of the International Computer Music Conference*, Cologne, Germany.

Prusinkiewicz, P. (1986). *Score generation with L-systems.* Ann Arbor, MI: MPublishing, University of Michigan Library.

Puckette, M. (1996). Pure Data. *Proceedings of the International Computer Music Conference*, 224-227.

Puckette, M., Apel, T., & Zicarelli, D. (1998). Real-time audio analysis tools for Pd and MSP. *Proceedings of the International Computer Music Conference*.

Purwins, H., Blankertz, B., & Obermayer, K. (2000). A new method for tracking modulations in tonal music in audio data format. *Proceedings of the International Joint Conference on Neural Networks, 6*, 270–275.

Reich, S. (1968). Music as a gradual process. In P. Hillier (Ed.), *Writings on music:* 1965-2000 (pp. 34-36). New York, NY: Oxford University Press (2002).

Reigle, R. (2008). Spectral musics old and new. *Proceedings of the Istanbul Spectral Music Conference*.

Ricard, J. (2004). *Towards computational morphological description of sound*. Master thesis, Pompeu Fabra University, Barcelona.

Roads, C. (1988). Introduction to granular synthesis. *Computer Music Journal*. The MIT Press, *12*(2), 11-13.

Roads, C. (1996). *Computer music tutorial*. Cambridge, MA: The MIT Press.

Roads, C. (1998). Micro-sound: History and illusion. *Proceedings of the Digital Audio Effects Workshop*.

Roads, C. (2001). *Microsound*. Cambridge, MA: The MIT Press.

Roads, C. (2006). The evolution of granular synthesis: An overview of current research. *Proceedings of International Symposium on the Creative and Scientific Legacies of Iannis Xenakis*.

Rowe, R. (1993). *Interactive music systems: Machine listening and composing*. Cambridge, MA: The MIT Press.

Rowe, R. (2001). *Machine musicianship*. Cambridge, MA: The MIT Press.

Rowe, R. (2009). Split levels: Symbolic to sub-symbolic interactive music systems. *Contemporary Music Review*, *28*(1), 31–42.

Savary, M., Schwarz, D., & Pellerin, D. (2012). DIRTI—Dirty Tangible Interfaces. *Proceedings of the New Interfaces for Musical Expression Conference*, 347-350.

Schaeffer, P. (1966). *Traité des objets musicaux*. Paris: Le Seuil.

Scheirer, E. (1998). Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, *103*(1), 588-601.

Schloss, W. (1985). *On the automatic transcription of percussive music: From acoustic signal to high-level analysis*. PhD dissertation, Stanford University.

Schnell, N., Cifuentes, M. A. S., & Lambert, J. P. (2010). First steps in relaxed real-time typo-morphological audio analysis/synthesis. *Proceedings of the Sound and Music Computing Conference*.

Schwarz, D. (2000). A system for data-driven concatenative sound synthesis. *Proceedings of the International Conference on Digital Audio Effects*, 97-102.

Schwarz, D. (2004). *Data-driven concatenative sound synthesis*. PhD Dissertation, Académie de Paris, Université Paris 6.

Schwarz, D. (2005). Current research on concatenative sound synthesis. *Proceedings of the International Computer Music Conference*.

Schwarz, D. (2006a). Real-time corpus-based concatenative synthesis with CataRT. P*roceedings of the International Conference on Digital Audio Effects*, 1-7.

Schwarz, D. (2006b). Concatenative sound synthesis: The early years. *Journal of New Music Research*, *35*(1), 3-22.

Schwarz, D., Britton, S., Cahen, R., & Goepfer, T. (2007). Musical applications of real-time corpus-based concatenative synthesis. *Proceedings of the International Computer Music Conference*, 47-50.

Schwarz, D., Cahen, R., & Britton, S. (2008). Principles and applications of interactive corpus-based concatenative synthesis. *Journées d'Informatique Musicale*.

Schwarz, D., & Schnell, N. (2009). Sound search by content-based navigation in large databases. *Proceedings of the Sound and Music Computing Conference*, 253-258.

Schwarz, D. (2012). The sound space as musical instrument: Playing corpus-based concatenative synthesis. *Proceedings of the New Interfaces for Musical Expression Conference*.

Schwarz, D., & Hackbarth, B. (2012). Navigating variation: Composing for audio mosaicing. *Proceedings of the International Computer Music Conference*.

Sell, G. (2010). Diffusion-based music analysis: A non-linear approach for visualization and interpretation of the geometry of music. PhD dissertation, Stanford University.

Serrà, J., Gómez, E., Herrera, P., & Serra, X. (2008). Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech and Language Processing*, *16*(6), 1138-1152.

Shlens, J. (2005). A tutorial on principal component analysis. Systems Neurobiology Laboratory, University of California at San Diego.

Sioros, G., & Guedes, C. (2011). Automatic rhythmic performance in Max/MSP: The kin. rhythmicator. *Proceedings of the International New Interfaces for Musical Expression Conference*.

Skočaj, D., Leonardis, A., & Bischof, H. (2007). Weighted and robust learning of subspace representations. *Pattern recognition*, *40*(5), 1556-1569.

Smalley, D. (1986). Spectro-morphology and structuring processes. In S. Emmerson (Ed.), *The language of electroacoustic music* (pp. 61-93). Basingstoke: Macmillan.

Smalley, D. (1997). Spectromorphology: Explaining sound-shapes. *Organised Sound*, *2*(2), 107-126.

Smalley, D. (1999). Etablissement de cadres relationnels pour l'analyse de la musique postschaefferienne. In D. Dufour & J.-C. Thomas (Eds.), *Ouïr, entendre, écouter, comprendre après Schaeffer* (pp. 177-213). Paris: INA/Buchet-Chastel.

Stoll, T. (2011). CBPSC: Corpus-based processing for SuperCollider. *Proceedings of the International Computer Music Conference*, Huddersfield, UK

Stone, H. (1972). *Introduction to computer organization and data structures*. New York, NY: McGraw-Hill.

Sturm, B. L. (2004). MATConcat: An application for exploring concatenative sound synthesis using MATLAB. *Proceedings of the International Conference on Digital Audio Effects*.

Sturm, B. (2006a). Concatenative sound synthesis and intellectual property: An analysis of the legal issues surrounding the synthesis of novel sounds from copyright-protected work. *Journal of New Music Research*, *35*(1), 23-33.

Sturm, B. (2006b). Adaptive concatenative sound synthesis and its application to micromontage composition. *Computer Music Journal*, *30*(4), 46-66.

Taube, H. (2004). *Notes from the metalevel: Introduction to algorithmic music composition*. London, UK: Taylor & Francis Group.

Taylor, C. (1988). Various approaches to and definitions of creativity. In R. Sternberg (Ed.) *The nature of creativity* (pp. 99.124). New York, NY: Cambridge University Press.

Taylor, R. (1990). Interpretation of the correlation coefficient: A basic review. *Journal of Diagnostic Medical Sonography*, *6*(1), 35-39.

Temperley, D. (1999). What's key for key? The Krumhansl-Schmuckler key finding algorithm reconsidered. *Music Perception*, *17*(1), 65–100.

Temperley, D. (2005). A bayesian key-finding model. *2005 MIREX Contest—Symbolic Key Finding*. Retrieved October 27, 2013, from: http://www.music-ir.org/evaluation/mirex-results/sym-key/index.html.

Terhardt, E. (1974). On the perception of periodic sound fluctuations (roughness). *Acustica*. *30*(4), 201–213.

Terhardt, E. (1984). The concept of musical consonance: A link between music and psychoacoustics. *Music Perception*, *1*(3), 276-295.

Thomas, J.-C. (1999). Introduction. In D. Dufour & J.-C. Thomas (Eds.), *Ouïr, entendre, écouter, comprendre après Schaeffer* (pp. 11-49). Paris: INA/Buchet-Chastel.

Thoresen, L. (1985). Un model d'analyse auditive. *Analyse Musicale*, *1*.

Thoresen, L. (1987). Auditive analysis of musical structures: A summary of analytical terms, graphical signs and definitions. *Proceedings from ICEM Conference on Electroacoustic Music*.

Thoresen, L. (2007a). Form-building transformations: An approach to the aural analysis of emergent musical forms. *The Journal of Music and Meaning*, *4*(3).

Thoresen, L. (2007b). Spectromorphological analysis of sound objects: An adaptation of Pierre Schaeffer's typomorphology. *Organised Sound, 12*(2), 129–141.

Truax, B. (1988). Real-time granular synthesis with a digital signal processor. *Computer Music Journal, 12*(2), 14-26.

Tzanetakis, G., & Cook, P. (1999). Multifeature audio segmentation for browsing and annotation. *Proceedings IEEE Workshop on applications of Signal Processing to Audio and Acoustics*.

Tzanetakis, G., & Cook, P. (2001). Automatic musical genre classification of audio signals. *Proceedings of the International Conference on Music Information Retrieval*.

Tzanetakis, G. (2002). *Manipulation, analysis, and retrieval systems for audio signals*. PhD dissertation, Princeton University.

Uhle, C., & Herre, J. (2003). Estimation of tempo, micro time and time signature from percussive music. *Proceedings of the Digital Audio Effects Workshop*.

Wang, G., Fiebrink, R., & Cook, P. (2007). Combining analysis and synthesis in the ChucK programming language. *Proceedings of the International Computer Music Conference*, 35-42.

Waters, S. (2000). Beyond the acousmatic: Hybrid tendencies in electroacoustic music. In S. Emmerson (Ed.), *Music, electronic media and culture* (pp. 56-83). Aldershot: Ashgate.

Wessel, D. (1979). Timbre space as a musical control structure. *Computer Music Journal*, *3*(2), 45-52.

Wishart, T. (1994). *Audible design: A plain and easy introduction to practical sound composition*. York, UK: Orpheus the Pantomime Ltd.

Wold, E., Blum, T., Keislar, D., & Wheaton, J. (1996). Content-based classification search and retrieval of audio. *IEEE Multimedia*, *3*(3), 27–36.

Xenakis, I. (1971). *Formalized music*. Bloomington, IN: Indiana University Press.

Xiang, P. (2002). A new scheme for real-time loop music production based on granular similarity and probability control. *Proceedings of the International Conference on Digital Audio Effects*, 89-92.

Zils, A., & Pachet, F. (2001). Musical mosaicing. *Proceedings of the International Conference on Digital Audio Effects*.

APPENDIXES

Appendix A

List of Representative Concatenative Sound Synthesis Software, Inspired by Schwarz (2006b) and Sturm (2006b)

NAME, AUTHOR (YEAR)	**Unit type**	**Audio representation**	**Selection process**	**Concatenation type**	**Code language/ software environment**	**Speed**
Caterpillar, Schwarz (2000)	Uniform and non-uniform	Low- and high-level audio features	Global and local constraints	Slight overlap with crossfade	Matlab	Offline
Musaicing, Zils & Pachet (2001)	Uniform	Low-level audio features	Global and local constraints	?	?	Offline
Soundmosaic, Hazel (2001)	Uniform	Waveform	Maximum inner product	Direct substitution	C++	Offline

Soundscapes, Hoskinson & Pai (2001)	Non-uniform	Wavelet transform representation	Random selection based on a probabilistic model of the audio source	Slight overlap with crossfade	Java	Online
Granuloop, Xiang (2002)	Uniform	Short-time Fourier transform	Random selection based on a probabilistic model of the audio source	Direct substitution combined with digital signal processing	Pure Data	Online
Sound Clustering Synthesis, Kobayashi (2003)	Uniform	Short-time Fourier transform	Local constraints	?	?	Offline
Directed Soundtrack Synthesis, Cardle et al. (2003)	Uniform	Low-level audio features	Local constraints	?	?	Offline
MoSievius, Lazier & Cook (2003)	Non-uniform	Low-level audio features	Local search	Overlap/add and PSOLA	C++	Online
Synful, Lindemann (2004)	Non-uniform	High-level audio features	Lookahead	Transformations on the selected units utilizing reconstructive phrase modeling	?	Online
MATConcat, Sturm (2004)	Uniform	Low-level audio features	Local search	User-defined, windowed	Matlab	Offline
Soundspotter, Casey (2005)	Uniform and non-uniform	Low-level audio features	Local and global constraints	?	C++	Online

Ringomatic, Aucouturier & Pachet, (2005)	Uniform	Low- and mid-level audio features	Local and global constraints	?	Java	Online
Audio Analogies, Simon *et al.* (2005)	Non-uniform	High-level audio features	Local and global constraints	Digital signal processing techniques	?	Online
Skeleton, Jehan (2005)	Uniform	Low- and mid-level audio features	Local constraints	Direct substitution	Objective-C	Offline
CataRT, Schwarz (2005)	Non-uniform	Low-level audio features	Local search, navigable	Units played at specific times with an amplitude envelope combined with digital signal processing	Max/MSP (uses FTM, Gabor, and MnM libraries)	Online
Vienna Symphonic Library, (2006)	Non-uniform	High-level audio features	Lookahead	?	?	Online
MEAPsoft, Weiss et al. (2009)	Non-uniform	Low-level audio features	Local search	?	Java	Offline
timbreID Brent (2009)	Uniform and non-uniform	Low-level audio features	Local search	(Not applicable)	Pure Data	Offline/Online
Audiogarden Frisson et al. (2010)	Non-uniform	Low-level audio features	?	?	MediaCycle framework	
AudioGuide Hackbarth (2010)	Non-uniform	Low-level audio features	Local search	Direct substitution	Python	Online

Appendix B

Related Publications

Bernardes, G., Davis, M. E. P., Guedes, C., & Pennycook, B. (2014). Considering roughness to describe and generate vertical musical structure in content-based algorithmic-assisted audio composition. *Proceedings of the Joint International Computer Music and Sound and Music Computing Conference*.

Bernardes, G (2014). Para além da recuperação de dados: Proposta de um modelo de geração automática de narrativas musicais dialógicas. *Workshop Narrativa, Média e Cognição*.

Bernardes, G., Guedes, C., & Pennycook, B. (2013). EarGram: An application for interactive exploration of concatenative sound synthesis in Pure Data. In M. Aramaki, M. Barthet, R. Kronland-Martinet, & S. Ystad (Eds.), *From sounds to music and emotions* (pp. 110-129). Berlin-Heidelberg: Springer-Verlag.

Bernardes, G., Peixoto de Pinho, N., Lourenço, S., Guedes, C., Pennycook, B., & Oña, E. (2012). The creative process behind *Dialogismos I*: Theoretical and technical considerations. *Proceedings of the ARTECH 2012—6th International Conference on Digital Arts*.

Bernardes, G., Guedes, C., & Pennycook, B. (2012). EarGram: An Application for interactive exploration of large databases of audio snippets for creative purposes. *Proceedings of the 9th International Symposium on Computer Music Modeling and Retrieval (CMMR)*.

Bernardes, G., Guedes, C., Pennycook, B. (2010). Style emulation of drum patterns by means of evolutionary methods and statistical analysis. *Proceedings of the Sound and Music Computing Conference*.

Appendix C

Chronological List of Works Created With EarGram

Title of the Work	Author	Type of Work	Performer(s) of the Premiere	Date of the Premiere	Event/Location of the Premiere
In Nuce	Ricardo Ribeiro	Piece for saxophone and live-electronics	Gilberto Bernardes	31-03-2011	Escola Superior de Música e Artes do Espetáculo do Porto, Portugal.
In Limine	Ricardo Ribeiro	Piece for ensemble and live-electronics	Sond'Ar-te Electric Ensemble	02-12-2011	Centro Cultural de Cascais, Portugal
Schizophonics	Rui Dias	Electroacoustic piece	Rui Dias (sound diffusion)	05-10-2012	Festival Manobras, Porto, Portugal
Dialogismos I	Nuno Peixoto	Piece for saxophone and electronics	Gilberto Bernardes	12-10-2012	ISMIR 2012 Conference, Maus Hábitos, Porto, Portugal
Dialogismos II	Nuno Peixoto	Piece for ensemble and electronics	Sond'Ar-te Electric Ensemble	10-10-2012	Goethe-Institut, Lisbon, Portugal
Your Feet	Nuno Peixoto	Piece for voice, percussion, and electronics	Rita Redshoes and Nuno Aroso	14-12-2012	Centro Cultural de Belém, Lisboa, Portugal
A Passos de Naufrágio	Nuno Peixoto	Piece for timbales and electronics	José Silva	30-6-2013	Escola Profissional de Música de Viana do Castelo
Urban Sonic Impression	Rui Dias and Gilberto Bernardes	Sound installation		28-10-2013	Sonoscopia, Future Places Festival, Porto, Portugal